#31
S. Bond
2/8/00

ATTORNEY DOCKET NO. N0003/7000

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

Applicant:	Glenn W. Hutton et al.
Serial No.:	08/533,115
Filed:	September 25, 1995
For:	POINT-TO-POINT INTERNET PROTOCOL
Examiner:	M. H. Rinehart
Art Unit:	2756

CERTIFICATE OF EXPRESS MAILING

"Express Mail" mailing label number: EL445948630US
Date of Deposit: **July 14, 1999**

I hereby certify that the following Correspondence is being deposited with the United States Postal Service "Express Mail Post Office to Addressee" service pursuant to 37 C.F.R. §1.10 on the date indicated above in an envelope addressed to Commissioner of Patents and Trademarks, BOX ISSUE FEE, Washington, D.C. 20231.

Frances M. Cunningham
Frances M. Cunningham

Assistant Commissioner for Patents
Washington, D.C. 20231

DECLARATION OF PRIOR INVENTION IN THE UNITED STATES TO OVERCOME CITED PATENT UNDER 37 CFR 1.131

Sir/Madam:

This declaration is to establish completion of the invention in this application in the United States at a date prior to May 23, 1995, the effective date of prior art patent 5,581,552, cited by the Examiner. The undersigned Declarant was added as a named Inventor in the above-identified patent application. The Declarant's statements set forth below establishes conception of the invention prior to the effective date of the reference coupled with due diligence from prior to the effective date of reference to filing of the application . Exhibit B is submitted herewith to support the Declarant's statements. This Declaration is submitted prior to final rejection or payment of the issue fee in the application.

1. I am the named inventor in the United States Patent Application 08/533,115, filed September 25, 1995, entitled "POINT-TO-POINT INTERNET PROTOCOL".

2. Prior to May of 1995, I, with other named inventor(s), jointly conceived of the subject matter disclosed in the above-identified patent application.

3. A number of weeks after the conception of the inventive subject matter and various refinements to the inventive concepts, I helped form, and became a principal of, the Internet Telephone Company, a Florida Corporation having a place of business at One South Ocean Boulevard, Suite 305, Boca Raton, Florida 33432.

4. Following formation of the Internet Telephone Company, a detailed design specification entitled "Internet Telephone Company WebPhone Design", a copy of which is attached hereto as Exhibit B, was generated to memorialize a product implementation of the inventive concepts and to provide the basis from which coding and testing of a working embodiment of the inventive concepts continued diligently until the filing date of this patent application, September 25, 1995.

I hereby declare that all statements made herein of my own knowledge are true and that all statements made on information and belief are believed to be true; and further that these statements were made with the knowledge that willful false statements and the like so made are punishable by fine or imprisonment, or both, under Section 1001 of Title 18 of the United States Code, and that such willful false statements may jeopardize the validity of the application or any patent issued thereon.

Craig B. Strickland

06/14/99
Date

Citizenship: Canada
Residence Address: 5713 NW 65th Terrace, Tamarac, Florida
Post Office Address: 5713 NW 65th Terrace, Tamarac, Florida

Internet Telephone Company

webPhone™ *Design*

[redacted]

One South Ocean Blvd.
Suite 305
Boca Raton
Florida
33432
Tel. 407.347.2447
Fax. 407-347-2445

THE INFORMATION CONTAINED HEREIN IS OF A HIGHLY CONFIDENTIAL AND PROPRIETARY NATURE AND IS NOT TO BE DISCLOSED TO ANYONE WITHOUT THE PRIOR WRITTEN CONSENT OF THE INTERNET TELEPHONE COMPANY.

EXHIBIT B

webPhone Structure and Function

The webPhone consists of a main window which looks and feels like a modern cellular flip phone and set of dialog boxes launched from the main window. See figure 1. The webPhone is controlled by clicking on objects (i.e. buttons, text and images) and dragging objects (i.e. lines, parties, messages, etc.).

The webPhone main window is 200x450 pixels closed and 200x590 pixels when the flip is opened. On a standard 640x480 display, when the user opens the flip door, the door detaches from the webPhone and is displayed on the side of the webPhone. This detached flip door is movable around the screen. When it is closed, it goes back onto the webPhone as before it was opened.

Buttons behave in one of two ways to the user. A button may be a *momentary* button which when pressed (left clicked on) gets pushed in then pops back out again or a button may be a toggle button which when pressed gets pushed in and stays in until pressed again (toggle buttons are either in a raised or depressed state). I will not make a joke here.

The objects comprising the webPhone main window are:

- display
- number pad
- line pad
- call function buttons
- phone function buttons
- audio control buttons and sliders

display

The display is 150x80 pixels and displays the following information:

party name

A text entry field using the READOUT truetype font. Text is 14 pixels high. The party name field can accomodate 20 to 25 characters on the display. If the user enters a name then presses [SND] to place the call and the user is not in the phone DIR, the *Directory Assistance* (Information) dialog will appear. If the user right clicks on the party name field, the *Update* phone DIR entry dialog will appear for that party if it exists thereby enabling the user to quickly modify the party's information.

When a call arrives, the caller's name will appear in the party name field as a caller ID feature.

party IP address

A text entry field using the READOUT truetype font. Text is 14 pixels high. To place a call to another user who has a known (fixed) IP address, the user enters the IP address in the party IP address field then presses [SND]. If the callee exists in the phone DIR and/or the call goes through, the callee's name will appear in the party name field (caller ID). If the IP address given is bad, the line status annunciator will say so.

WebPhone status annunciators

The 3rd line of the *webPhone* display is used to display iconic annunciators providing feedback to the user about the status of events taking place in the *webPhone*. The status annunciators are:

1. user is camped on one or more parties
2. default call forwarding is enabled (effects all parties with no specified call forwarding party)
3. call blocking is enabled (effects parties with call blocking enabled)
4. do not disturb is enabled
5. priority ringing is enabled (effects parties with priority ring enabled)
6. file transfer is occuring
7. voice mail transfer is occuring

Line number annunciator

Cycle through all lines by single clicking on the *Line number annunciator* (Li), the main LED or the line status annunciation text. The main LED color and line state annunciation text will change to reflect the state of the selected line. If the user is on a line with an active call, the *Line number annunciator* will return to reflect that line's status after a time out of 5 seconds. If no lines exist with active calls or no line is selected, the *Line number annunciator* will remain on the line which was last seleted (i.e. no time out occurs to change the Line number annunciator back).

local time/party's time

When there are no lines with active calls, the *webPhone* displays the current local time. When the user is on a line with an active call, the *webPhone* displays the remote party's time. By single clicking on the time, the user can cycle through the two different times as follows:

```
local time --> party's time
      ^              |
      |              |
      +--------------+
```

As the user changes lines, the time displayed will reflect the time format which was last selected for the selected line.

new vmail msgs/total vmail msgs

The *webPhone* displays the current number of new voice mail messages and the total number of voice mail messages as follows:

new / total

If the user single clicks on the *vmail msgs annunciator*, he/she can display the total number of voice mail messages. If the user single clicks on the *vmail msgs annunciator* again, it will revert back to display the current number of new voice mail messages. The *vmail msgs annunciator* will automatically revert back to display the number of new voice mail messages after 5 seconds.

call duration

The duration of the current call is displayed in mm:ss format. As the user cycles through the lines by clicking on the Line number annunciator, the call duration annunciator changes to reflect that line's call duration if any.

main led

This LED mirrors the LED of the currently selected line. The LED colors are specified in figure 48. The colors represent the state of the call on the selected line.

line status text

Informs the user as to the state of the currently selected line. See figure 48.

list arrow

Enables the user to pop down the list of parties on the selected conference call.

Conference party list

When a user selects a active line with a conference call, the name of the first party on the conference call is displayed in the *party name* field in the display along with the *list arrow* described above. Once the user presses the list arrow to obtain the conference party list, the user can view all the parties present on the conference call (even those parties added to the conference by another party on the conference call).

If the user right clicks on an unselected line with a conference call (i.e. while engaged on another active line), the conference party list is displayed (no need to press the list arrow) for viewing and manipulation of the parties as described below. In the event the user does nothing with the list for 5 seconds or another object is selected (e.g. another button is pressed), the display will revert back to displaying the information about the currently selected line.

The user may remove one or more parties from the conference call by selecting them in the conference party list and pressing [END]. The user may also transfer one or more parties from the conference call by selecting them and dragging them to a free (IDLE) line. If the user is placed on hold by a party on the conference call, the only way the user may know this is to view the conference party list and check the face icon of the parties in the list.

Priority ring party list

When the user enables priority ringing (depresses [PRI]) or right clicks anytime on [PRI], a list of parties who have priority ringing enabled will appear in the display. The user may disable priority ringing for one or more parties by selecting them in the list and pressing the |**Delete**| key. This removes the parties from the priority ring list and updates the effected parties' records in the phone directory by disabling priority ringing. The user may also disable priority ringing for one or more parties by updating their records directly in the phone directory. In the event the user does nothing with the list for 5 seconds or another object is selected (e.g. another button is pressed), the display will revert back to displaying the information about the currently selected line. If there are no parties with priority ringing enabled, pressing [PRI] does nothing.

Call blocking party list

When the user enables call blocking (depresses [BLK]) or right clicks anytime on [BLK], a list of parties who have call blocking enabled will appear in the display. The user may disable call blocking for one or more parties by selecting them in the list and pressing the |**Delete**| key. This removes the parties from the call blocking list and updates the effected parties' records in the phone directory by disabling call blocking. The user may also disable call blocking for one or more parties by updating their records directly in the phone directory. In the event the user does nothing with the list for 5 seconds or another object is selected (e.g. another button is pressed), the display will revert back to displaying the information about the currently selected line. If there are no parties with call blocking enabled, pressing [BLK] does nothing.

In order to change the action to be performed when an inbound call arrives from a party with call blocking enabled (i.e. reject the call or give them to the answering machine), the user must update that party's record directly in the phone directory.

Camped on party list

When the user right clicks on [CMP], the camped on party list appears in the display. The user may remove a camp on a party by selecting the party and pressing the |**Delete**| key. In the event the user does nothing with the list for 5 seconds or another object is selected (e.g. another button is pressed), the display will revert back tc displaying the information about the currently selected line.

speed dial info

When a user right clicks on [0] or [1] or ... or [9], the name, alias, e-mail address and IP address (if known) of the party assigned to that speed dial position will appear in the display for 5 seconds or until another object is selected (e.g. another button is pressed), whichever comes first, then the display will revert back to displaying the information about the currently selected line.

line info

When a user right clicks on [L1], [L2], [L3] or [L4], the name, alias, e-mail address and IP address (if known) of the party on that line will appear in the display for 5 seconds or until another object is selected (e.g. another button is pressed), whichever comes first, then the display will revert back to displaying the information about the currently selected line.

FWD party list

When the user enables call forwarding (depresses [FWD]) or right clicks anytime on [FWD], a list of parties who have call forwarding enabled will appear in the display. The user may disable call forwarding for one or more parties by selecting them in the list and pressing the |**Delete**| key. This removes the parties from the call forwarding list and updates the effected parties' records in the phone directory by disabling call forwarding. The user may also disable call forwarding for one or more parties by updating their records directly in the phone directory. In the event the user does nothing with the list for 5 seconds or another object is selected (e.g. another button is pressed), the display will revert back to displaying the information about the currently selected line. If there are no parties with call forwarding enabled, pressing [FWD] does nothing.

In order to change a party's forwarding party (i.e. the party to be called) when an inbound call arrives from a party with call forwarding enabled, the user must update that party's record directly in the phone directory.

<u>number pad</u>

[0],[1],..[9] & [.]

The number buttons are 34x26 pixels. The number buttons may be used to enter a party's IP address. To erase an incorrect entry, the user must use the |**Backspace**| key on the keyboard. The number buttons also house the ten speed dial positions. The user may assign a party to one of the ten number buttons then initiate a speed dial by simply pressing [n] then [SND]. If the user right clicks on [n], the information about the party who is assigned to that speed dial position will be displayed.

<u>line pad</u>

[L1 o], [L2 o], [L3 o] & [L4 o]

The line buttons are 46x26 pixels. Line buttons are toggle buttons. Each line button has a letter and number indicating which line it is and a led which indicates the state of the call on that line (see figure

48). When a line has a call with more than one party (conference call), the line button will replace the letter L with the letter C indicating that it contains a conference call. When a line containing a conference call reverts back to having only one party on the call, the line button will replace the letter C with the letter L indicating that it now contains a regular call. The line buttons work like the buttons on your car radio, only one can be depressed at a time. When a line button is depressed it is pre-selected or the active line. Pressing a line button when no inbound calls exist pre-selects that line for the next inbound or outbound call (depresses the line button). Pressing a line button when an inbound call arrives on that line answers the call (depresses the line button). Pressing a line button when the line is IN USE places the call on hold (raises the line button). Pressing a line button when the line is on hold takes the line off hold (depresses the line button).

call function buttons

The call function buttons are 46x26 pixels.

[RCL]

Recall last number. [RCL] is a momentary button. Pressing [RCL] recalls the last party called by displaying the party's name, alias, e-mail address and IP address (if known), selecting a free line (if a line has not already been pre-selected) then automatically pressing [SND] to initiate the call. The user may also right click on [RCL] to view the party's name, alias, e-mail address and IP address (if known) in the display. If the user does not press [SND] to intiate the call within 5 seconds from right clicking on [RCL], the display will revert back to displaying the information about the currently selected line. If the user presses [RCL] while engaged on an active line, that line will be placed on hold just as if the user had pressed [HLD] or deselected that line. If no free lines are available, pressing [RCL] will do nothing, however right clicking on [RCL] will still display the information about the last party called.

[END]

Terminates a call. [END] is a momentary button. If the user presses [END] when no lines are active no action is performed.

[SND]

Places and answers a call. [SND] is a momentary button. If the user presses [SND] when there are no free lines available or no party name is present in the *party name* field in the display or no inbound calls exist then no action is performed. When a call is placed or answered, the status of the call is indicated in the display and in the led color on the line with the active call.

[DND]

Do not disturb. [DND] is a toggle button. When [DND] is depressed it instructs the webPhone not to disturb the user with inbound calls but to send all inbound calls to the answering machine. When do not disturb is enabled, the display will annunciate the *do not disturb* icon. To tu.:n off do not disturb, the user presses the depressed [DND] button.

[MUT]

Mute the conversation. [MUT] is a toggle button. When the user presses [MUT] the party on the call or all parties on a conference call can not hear the user (the microphone is effectively disabled). When mute is enabled, the *main led* and *line status text* in the display and the led color on the line button change to indicate that the call is muted. If the user presses [MUT] when no lines are selected or the selected line is in a state that cannot accept muting no action is performed. If a party mutes the call, the user has no indication of it. To unmute a call, the user presses the depressed [MUT] button.

[HLD]

Places the call on hold. [HLD] is a momentary button. When the user presses [HLD] the party on the call or all parties on a conference call are placed on hold (the microphone and speaker are effectively disabled). When hold is enabled, the *main led* and *line status text* in the display and the led color on the line button change to indicate that the call is on hold. If the user presses [HLD] when no lines are selected or the selected line is in a state that cannot accept muting no action is performed. If a party place the call on hold, the *main led* and *line status text* in the display and the led color on the line button change to indicate that the call has been placed on hold by the party. To take a call off hold, the user must press the line button of the holding call.

[CMP]

Camps on a party. [CMP] is a momentary button. Camping on a party serves to ensure that your call to that party will go through when the party is available (no longer busy or is back online). Think if it as a perpetual redial. When the user calls a party and the party is either BUSY or OFFLINE, the user may press [CMP] to camp on that party. To remove a camp on a party, the user must first display the camp list by right clicking on [CMP] then select the desired party and press the |**delete**| key.

[BLK]

Enables or disables call blocking. [BLK] is a toggle button. When call blocking is enabled (button is depressed) all inbound calls from parties who have call blocking enabled will be either rejected or given to the answering machine thereby not disturbing the user. Whether the call is rejected or given to the answering machine is specified in each party's record in the phone directory. If the call is rejected, the party will see REJECTED in their display.

[PRI]

Enables or disables priority ringing. [PRI] is a toggle button. When priority ringing is enabled (button is depressed) all inbound calls from parties who have priority ringing enabled will generate the priority ring sound effect when the call arrives. Priority ringing is specified in each party's record in the phone directory.

[FWD]

Enables or disables call forwarding. [FWD] is a toggle button. When call forwarding is enabled (button is depressed) all inbound calls from parties who have call forwarding enabled will cause the *webPhone* to route the call to the designated call forwarding party as specified in the party's record in the phone directory. If the calling party has not been assigned a call forwarding party and call forwarding is enabled, the call will be routed to the default call forwarding party (assigned to [FWD] itself) if it exists. To assign a default call forwarding party the user drags the desired party from either the phone directory, line, camp list, speed dial position, etc. and drops it on [FWD].

<u>phone function buttons</u>

The phone function buttons are 46x26 pixels.

[?]

Help. [HLP] is a momentary button. Pressing [?] launches the *webPhone* manual - wpman.exe, an interactive, multimedia tutorial and help system. Puts the user right at the start of the manual.

[CFG]

Configure the *webPhone*. [CFG] is a momentary button. Pressing [CFG] lauches the *configuration* dialog which enables the user to change the operating parameters of the *webPhone*. See figures 16 - 22.

[DIR]

Phone directory. [DIR] is a momentary button. Pressing [DIR] lauches the *phone directory* dialog which enables the user to add, update, sort, view and delete parties and obtain directory assistance. See figures 7 - 10.

[MSG]

Voice mail messages. [MSG] is a momentary button. Pressing [MSG] lauches the *voice mail messages* dialog which enables the user to view, sort, playback, delete, save and restore voice mail messages as well as create, playback, delete, save, restore custom outgoing messages and assign them to parties in the phone directory. See figures 4 - 6.

[DAT]

Data file transfer. [DAT] is a momentary button. Pressing [DAT] lauches the *data file transfer* dialog which enables the user to monitor and control the progress of data file transfered to and from parties. This is also the dialog which enables users to retrieve their e-mail and create and send e-mail. See figures 13 - 15.

[LOG]

Call activity log. [LOG] is a momentary button. Pressing [LOG] lauches the *call activity log* dialog which enables the user to view, sort, search for, print and delete call related events. The user may initiate a call to a party by double clicking on an event. See figures 11 - 12.

X

Exits the *webPhone*. If the user has one or more active calls, an information dialog (see figure 2.) will appear asking the user if he/she wishes to really exit and terminate the active calls.

[]

Minimizes or iconifies the *webPhone*. The *webPhone* icon will display the main LED color for the currently selected line.

webPhone

This is the *webPhone* about text button. When pressed the user obtains the *About* dialog. See figure 3.

<u>audio control buttons and sliders</u>

Control the recording and playback of voice mail and outgoing messages. Operate exactly like conventional audio tape deck controls.

flip open/close

Opens and closes the flip door

progress bar

Displays the extent of progress during playback and recording of audio. Recorded voice mail is limited to 2 minutes and recorded outgoing messages is limited to 30 seconds. These parameters are currently not configurable by the user (via [CFG]) - should we allow the user to change these parameters?

[|<]

Rewinds the tape to the beginning. *[|<]* is a momentary button.

[>|]

Fast forwards the tape to the end of the recording. *[>|]* is a momentary button.

[x]

Aborts recording or playback operation. *[x]* is a momentary button. If the user is recording a voice mail message and decides not to deliver it, s/he would press *[x]* to abort the recording then press [END] to terminate the call without sending voice mail.

[<|]

Sound card speaker. *[<|]* is a toggle button. Plays back audio on the sound card's speaker. *[<|]* is only active (not dimmed) when the user has bóth a sound card and the IPC.

[>]

Plays back audio. *[>]* is a special type of momentary button. When pressed it starts playing audio and pops out to become the Pause button *[||]*. When *[||]* is pressed it pauses playback of the audio and pops out to become *[>]* again.

[•]

Stops playback of audio. *[•]* is a momentary button.

[o]

Records audio. *[o]* is a toggle button. When *[o]* is depressed the user is in record audio and can record voice mail or an outgoing message. To stop recording, the user may press *[o]* again or press *[•]*.

SPK slider

Speaker volume control. Enables the user to adjust the output volume of the audio received during conversation and playback of voice mail and outgoing messages. If the user does not have the IPC, the SPK control attenuates the sound card's speaker volume, If the user has the IPC, the SPK control attenuates the IPC's speaker volume and the sound card's speaker volume must be attenuated via the sound card's audio control panel.

MIC slider

Microphone volume control. Enables the user to adjust the input volume of the audio recorded during conversation and recording of voice mail and outgoing messages. If the user does not have the IPC, the MIC control attenuates the sound card's microphone volume, If the user has the IPC, the SPK control attenuates the IPC's microphone volume and the sound card's microphone volume must be attenuated via the sound card's audio control panel.

The ITEL operator's have a [USR] button on their webPhone to acquire a user's *webphone.cfg* file during registration.

Implementation

The webPhone will be developed under MS Windows using Borland C++ v.4.51. This compiler was chosen because of its extensive class library, the existence of C++ templates, OLE 2.0 support and familiarity by the programming staff.

Platform

The *webPhone* will initially exist as a 16 bit version. A 32 bit version will be released later due to porting of socket and MCI code from 16 bit to 32 bit. Both versions will be capable of running on MS Windows 3.x and above. The 32 bit version will require the win32s subsystem to run on MS Windows 3.x (we will provide the user with the ability to obtain the win32s subsystem from the Itel Home Page).

The *webPhone* will use the w_char character set to allow for future portability to foreign languages based upon 16 bit characters such as kanji, arabic, hebrew, etc.

The 32 bit version will employ threads where necessary to improve performance in the PhoneManager (PM) and its AudioEngines (AE).

Architecture

The *webPhone* consists of a Graphic User Interface (GUI), a User Interface control (UI), a PhoneManager (PM) and an AudioEngine (AE). The GUI may be replaced by other GUI's such as X-Windows (UNIX), Presentation Manager (OS/2 Warp) and Macintosh without changing the UI,PM and AE to provide for fast porting to these other platforms. In addition, the *webPhone* can interface with multiple AEs to support a variety of audio compression and decompression algorithms (codecs) in software and hardware. For example, the *webPhone* interfaces with the software based GSM and TrueSpeech audio codecs via one AE (aesac.dll) and will interface with the ITEL phone card (IPC) via another AE (aeipc.dll). The *webPhone* will use the appropriate AE as required.

Refer to the System Architecture diagrams in figure 28 and the Software Architecture diagram in figure 29 for more details.

Objects

The GUI, UI, PM and AE use a number of objects to house and manipulate the data associated with the operation of the *webPhone*.

The GUI objects control the look and feel of the graphic user interface controls seen by the user which constitute the *webPhone*'s user interface. Some of the UI objects maintain and manage the many states of the *webPhone* and control the behavior of the graphic user interface controls. Refer to figures 40 - 46 for more details on GUI and UI objects.

The following objects are used primarily by UI and PM to manage the state of calls and tasks that are to be performed:

- line
- job
- party
- task

The AE only sees tasks. Refer to figure 47 for more details.

User Interface (UI)

The Seperation of GUI and UI Logic

Each Phone Control has two objects associated with it. The GUI Part is windowing system specific and the UI part is generic. When the GUI Control's state is change by the user it first checks with the UI to make sure it's OK to make the change.

UIControls and and their parents

A UIControl is always a child of UICollection. When the UIControl's sibling, the GUIControl, asks the UICrontrol if its OK to make a change, and this change request is accepted, the GUIControl still must ask its parent if the change is valid. The parent UICollection may have its own parent, another UICollection, that it must ask if the new value is OK.

The Grandparent of them all, the UIPhone

The UIPone is a UICollection. It has final say in all changes. It also must tell its children when the Phone Manager changes the phone state. It also creates jobs for the phone manager based on user actions. The UIPhone contains the following, the UILine Collection, all UIPopup collections, the MSG, DIR, LOG, CFG, DAT, PRI, BLK, and FWD buttons.

UILine

The UILine Collection contains all the collections and phone buttons that relate to the changes in the state of the line. Specifically, these are the four line buttons (e.g. L1, L2, L3, and L4), the RCL button, the HLD button, the MUT button, and the UICall Collection. The UIPhone is the parent of UILine.

UICall

The UICall Collection contains all the buttons related to calls. Specificly these are the number buttons, 0 - 9, ., the SND button, and the CMP button. The UICall's parent is the UILIne.

Windows Drag Drop

The DragObject function implements the server component of the drag and drop. A drag and drop server calls this function in response to a user initiated drag. Below is the function proto-type.

```
DWORD FAR PASCAL DragObject (
        HWND        Scope,        // Scope of drag
        HWND        Source        // Window handle initiating Drag
        WORD        Type,         // Dragged object type
        WORD        OfStruct,     // Handle to OFSTRUCT (not required)
        NPSTR       Data,         // Near point to drop data
        HCURSOR     Cursor,       // Handle to cursor
);
```

The Scope parameter limits the windows that can receive the drop. Only that window and its children will get the drop request. By setting it to GetDeskTopWindow(), any window can get the drop. The Source parameter is the server's window handle. The Type is the type of drag. Windows has four standard drag types (See table below). A drag is limited to a single application unless the Type parameter is or'ed with DRAGOBJ_EXTERNAL (0x0001).

If the object being dragged is a single file a OFSTRUCT global memory handle may be specified. But is not required, and may be set to NULL. Data points to a string containing the object data. Cursor is a cursor handle that shows when the mouse is over a window that will accept this type of drop.

As the user drags the object the function sends WM_QUERYDROPOBJECT to the window under the mouse. As long as the underlying window returns 0, the no entry cursor is displayed. If 1 is returned the cursor specified in the cursor parameter is displayed.

If the mouse left button is released over a window that will not accept the drop, the function returns 0, otherwise it returns non-zero. At this point the server builds a DROPINFO struct in global memory and sends it as the LPARAM in a WM_DROPFILES message.

```
typedef struct {
        WORD  DataOffset;      // Offset of the character data
        WORD  x;               // mouse x position of drop
        WORD  y;               // mouse y position of drop
        BOOL  InClient;        // True if in client area of window
        char  Data[n]          // Drop string data
} DROPINFO, FAR *LPDROPINFO;
```

WebPhone Drag Drop

The WebPhone drag and drop will use the standard windows drag and drop by adding some of its own object types (See table below). Each UIControl and GUIControl will have two member functions added to them (e.g. SetDragType(uint Type = 0) and AcceptDropTypes(uint Count = 0, uint* Types = NULL)). The SetDragType call will set the type of drag that the control will do if the mouse is moved out of the controls window with the left mouse button down. If the type is 0 no drags will happen. The AcceptDropTypes function will set the types of drags the control will accept. If either Count or Types is zero no drops will be accepted. (NOTE: since messages an ogms can be dragged to the file manager this drag will be of type DRAGOBJ_DATA)

Windows Standard Drag Types	**Value**	**Data**
DRAGOBJ_PROGRAM	0x0001	File Name
DRAGOBJ_DATA	0x0002	File Name
DRAGOBJ_DIRECTORY	0x0003	Directory Names

DRAGOBJ_MULTIPLE	0x0004	File and Directory Names Separated by spaces

WebPhone Drag Types

DRAGOBJ_CALL	0x0005	String with Job Pointer
DRAGOBJ_CONFERENCECALL	0x0006	String with Job Pointer
DRAGOBJ_DIRENTRY	0x0007	String with key for entry into phonedir.db

PhoneManager (PM)

The PM is a state machine. It consists of an array of pointers to functions and states which makes up an state-event table. When an event occurs (caused by the mouse, keyboard, mic, speaker or socket), it is up to the UI to determine if the event requires the attention of the PM. The PM is not notified of those events which only effect the GUI (e.g. user presses [DIR] to open the Phone Directory dialog). When the PM is to be notified of a call related event, the UI makes the following calls to PM where l = current lineID of call:

```
// trigger PM to perform action based upon event and current state
(*PhoneManager[line[l]]->job.state][event].fxn)(arg1,arg2,arg3);

// obtain new job state from PM
line[l]->job.state = PhoneManager[line[l]->job.state][event].newstate;
```

When the PM is to be notified of a non-call related event, the UI makes the following calls to PM:

```
// trigger PM to perform action based upon event and current job state
(*PhoneManager[job.state][event].fxn)(arg1,arg2,arg3);

// obtain new job state from PM
job.state = PhoneManager[~~line[l]->~~job.state][event].newstate;
```

Refer to the UI Triggered PM Events diagram in figure 53 for more details.

AudioEngine (AE)

Crippled WebPhone

Users may obtain a crippled version of the *webPhone* for trial use, at no cost, from the ITEL Home Page. The *webPhone* will become uncrippled once the user registers the *webPhone* (i.e. pays the $49.95 or $149.95). The *webPhone* will be crippled as follows:

Limited Functionality

The following buttons on the *webPhone* are active, the remainder are dimmed and inactive:

L1, L2, LOG, MSG, DIR, MUT, HLD, RCL, END, SND & CFG

Limited Talk Time

Allow only 60 seconds of conversation per call. The 60 seconds begins once the call is in the INUSE state. After the 60 seconds has elapsed, the call is disconnected and the calling user's *webPhone* will play the sound file *cnvtime.wav* which says in a female operator's voice something like this: "You must register your webphone for unlimited talk time".

3

Limited Phone Directory

Allow only 2 phone directory entries plus the ITEL phone directory entry. When the user attempts to add a forth phone directory entry the *webPhone* will play the sound file *cdiradd.wav* which says in a female operator's voice something like this: "You must register your webphone to have unlimited phone directory entries".

2

Limited Voice Mail

Allow only 1 functional voice mail message at any given time and restrict the message duration to 15 seconds for both incomming and outgoing messages. All other voice mail messages received will be displayed as dimmed audio cassette icons in the Voice Mail Messages dialog. This will enable users to still see the voice mail they have received in leu of the limitation. In the event the user attempts to play back a dimmed voice mail message, the *webPhone* will play the sound file *cvmlmsg.wav* which says in a female operator's voice something like this: "You must register your webphone for unlimited voice mail". The user may only delete dimmed voice mail messages and not copy or move them to a directory in the MS Window's file Manager.

Limited Conference Calling

The user is permitted only 1 conference call with a maximum of 2 remote parties on the conference. In the event the user attempts to add a third party to the conference, the *webPhone* will play the sound file *ccnfadd.wav* which says in a female operator's voice something like this: "You must register your webphone for unrestricted conferencing". If a remote party with a registered *webPhone* adds a third party to the conference (relative to the local user with the crippled *webPhone*), the user will not be allowed to converse with that party yet will be able to see that party in the conference list (a teaser).

No Spd?

Limited Speed Dail Position

Allows the user the first 2 speed dial positions: [1] and [2]. When the user attempts to add a party to any of the other 8 speed dial positions the *webPhone* will play the sound file *cspdadd.wav* which says in a female operator's voice something like this: "You must register your webphone for unrestricted speed dialing".

Will reject. when

Limited Activity Logging

Allows the user to view a maximum of 3 events in the Call Activity dialog. The call activity log activity.log will still retain the logged call activity. The user will still be able to see the total number of events that were logged (a teaser).

Limited Outgoing Messages

Allows the user only ~~one~~ 1 custom OGM. When the user attempts to add a second OGM the *webPhone* will play the sound file *cogmadd.wav* which says in a female operator's voice something like this: "You must register your webphone to define unlimited outgoing messages".

WebPhone Acquisition and Setup

When the *webPhone* is obtained from ITEL's Home Page:

The ITEL Home Page will enable the user to acquire the crippled version of the *webPhone* via ftp. All the installation files will be placed in a self extracting ZIP file named *itelwp10.exe*. The user will obtain the *itelwp10.exe* file and a *readme.txt* file which explains how to extract the installation files from the zip file into a temporary installation directory. Once extracted into a temporary directory, the *wpsetup.exe* file can be executed from MS Windows to install the *webPhone*.

When the *webPhone* is obtained from the purchase of the ITEL phone card:

The *webPhone* software will probably reside on two 3.5" 1.44MB floppy disks. The user will insert the floppy disk labeled "installation disk" and execute *wpsetup.exe* from MS Windows to install the *webPhone*.

Installation

InstallShield by Stirling will be used to develop the installation file *setup.exe* and create the installation image (to be zipped up into *itelwp10.exe* or placed on floppy diskettes). *Wpsetup.exe* will perform the following actions:

1. present the user with an attractive installation screen in a window
2. check for adequate disk space. If not enough disk space, inform user and exit setup.
3. present the user with a dialog box allowing the user to select:

- ( ) complete installation
- ( ) custom installation
- ( ) uninstall
- ( ) exit

Note: ( ) are radio buttons.

The following pertains to both complete and custom installation (if "install the *webPhone*" was selected):

4. search for previous version. If not found, say nothing to the user and continue. If found, ask the user if he/she would like to update or re-install. If update is selected, skip steps 6 thru 10 below (unless new or updated db files are required). If re-install is selected, continue with step 5 below.
5. prompt the user for a desired installation directory
6. prompt the user to complete the user information form thereby supplying his/her name, addr, phone, etc. and his/her e-mail address and IP address if known.
7. create the *webPhone* directory structure and install the files.
8. prompt the user to specify which existing program manager group or the name of a new group to place the webphone.exe, wpvmplay.exe, wpman.exe and the readme.wri icons into.
9. initialize database files
10. initialize counters in *webphone.cfg*
11. auto-recognize the ITEL phone card, if any, and test for operability; inform the user of the results and update *webphone.cfg* indicating the ITEL phone card is present and its version information. This is also performed every time *webphone.exe* is executed except the user is not notified of the results.
12. auto-recognize the user's sound card, if any, and test for compatibility; inform the user of the results and update *webphone.cfg* indicating a sound card is present and its name. This is also performed every time *webphone.exe* is executed except the user is not notified of the results.
13. associate audio files **.wpm* with *wpvmplay.exe* in win.ini (may not be necessary in Win95)
14. display "How to order" information
15. ask user if he/she would like to run the tutorial (wpman.exe).
16. inform the user installation was complete.

If custom installation was selected, the user would get the following dialog:

- [ ] install the *webPhone*
- [ ] define user information
- [ ] install database files >>

Note: [ ] are check boxes and >> is a "more" button

If the user selects "install the *webPhone*", he/she will follow steps 4 thru 16 above.

If the user selects "define user information", he/she will be prompted to complete the user information dialog (step 6. above) which will update *webphone.cfg*.

If the user selects "install database files", he/she will get another dialog prompting the user to select which database files to install:

- () configuration database - .webphone.cfg
- () phone directory - phonedir.db
- ·() voice mail messages directory - messages.dir
- () file transfer directory - files.dir
- () outgoing messages directory - ogm.dir
- () call activity log - activity.log

If any of the database tables are selected, those database tables will be re-created and initialized. In the event the "configuration database - webphone.cfg" is selected, the user will be prompted to enter his/her user information as if he/she had selected "define user information" in the custom installation dialog and steps 10-12 above will be performed.

E-mail Communication Protocol

Incomming messages

The following e-mail messages are transmitted to a remote user's Post Office Protocol (POP) server via the Simple Mail Transport Protocol (SMTP) using MIME by the PhoneManager (PM):

- Connect Request
- Camp Request
- Voice Mail
- File Transfer
- E-mail

Outgoing messages

The following e-mail messages are received from the local user's POP server vi the POP protocol using MIME by the PM:

- Connect Request
- Camp Request
- Voice Mail
- File Transfer
- E-mail
- Registration

Message structure

The e-mail messages are identified by unique subject information as described below:

!~~ITEL~~(L),~~TYPE~~,SID,EMAILADDR,IPADDR, PartNum, Total Parts
NETSPEAK

where

SID is the unique session identifier as an ulong in hex: 00000000-FFFFFFFF
EMAILADCR is the e-mail address of the sender: username@host.domain.org
IPADDR is the IP address of the sender as a char string: 198.98.127.9
PartNum is The file number of TotalParts

Total Parts is number of emails for this Type (1 of 10)

Message	L	TYPE
Connect Request	C	CALL
Camp Request	P	CAMPCALL
Voice Mail	V	VMAIL
File Transfer	F	FILEXFR
E-mail	M	EMAIL
Webphone Registration	R	REGISTRATION

Those messages which contain attached data (VMAIL, FILEXFR, EMAIL* and REGISTRATION) use the MIME protocol. VMAIL is in compressed wpm format (either GSM or Truespeech compressed file detectable by a magic cookie present in the file header).

* EMAIL may or may not contain attached data files

Note: the subject does not contain any non-printable ascii characters.

All messsages except EMAIL contain a text message in the message body in case the user's e-mail program (e.g. Eudora) captures the ITEL messages.

The text for a CALL or CAMPCALL message may say something like this:

> "You have a *webPhone* call from **name** at **emailAddr**. If you do not have a *webPhone* and wish to talk to **name**, contact the Internet Telephone Company at http://www.itel.com or call 800-NNN-ITEL."

where **name** and **emailAddr** are the full name and email address of the caller.

The text for a VMAIL message may say something like this:

> "You have *webPhone* voice mail from **name** at **emailAddr**. If you do not have a *webPhone* and wish to listen to your voice mail from **name**, contact the Internet Telephone Company at http://www.itel.com or call 800-NNN-ITEL."

The text for a REGISTRATION message may say something like this:

> "Attached is your *webPhone* registration file. Please save it as "webphone.reg" in your webphone directory to enable your *webPhone*. If you should encounter a problem with your *webPhone*, e-mail us at info@itel.com or call 800-NNN-ITEL. Thank you for purchasing the ITEL *webPhone*."

The text for a FILEXFR message may say something like this:

> "Attached is one or more files sent to you by **name** at **emailAddr** via their *webPhone*. If you do not have a *webPhone* and wish to easily perform file transfer over the net not to mention converse in real time, send and receive voice mail, etc., contact the Internet Telephone Company at http://www.itel.com or call 800-NNN-ITEL."

File System

Figure 32 represents the *webPhone* file system as it would exist on a user's hard disk. The following files are present:

in webphone\
readme.wri (MS Windows Write file describing how to install, list of files...)
webphone.exe (the *webPhone*)
wpvmlplay.exe (webphone vmail player associated with *.wpm files)
wpman.exe (authorware based tutorial, manual and help system)
webphone.reg (exists for sound card version after user registers)
wpsetup.exe (webphone installation and setup program)
activity.log (call activity log)
phonedir.db (phone directory database)
wpnet.dll (internet communications library)
wpaesac.dll (audio engine for audio card based webphone)
wpaeipc.dll (audio engine for ITEL phone card)
wpsac.dll (software based audio codec library - GSM and Truespeech)
wpipc.dll (ITEL phone card interface library - API)
ctpwin.dll (c-tree plus windows database interface library)
*.vbx (if any - we will try not to use any)

in webphone\vmail
messages.dir (directory of resident messages)

in webphone\vmail\in
XXXXXXXX.wpm (received compressed voice mail message files, X = 0-9)

in webphone\vmail\out
XXXXXXXX.wpm (sent compressed voice mail message files, X = 0-9)

in webphone\ogm
ogm.dir (directory of resident outgoing messages)
wpogm.wav (default outgoing message)
XXXXXXXX.wav (outgoing voice message files, X = 0-9)

in webphone\files
files.dir (directory of resident messages)

in webphone\files\in
. (received e-mail, executable, text, data and winapp files)

in webphone\files\out
. (transmitted e-mail, executable, text, data and winapp files)

in webphone\sounds
noanswer.wav ("the party does not answer")
offline.wav ("the party is not online")
busy.wav ("I'm sorry, the party is busy, please try again later")
ringout.wav (standard ring when calling)
ringin.wav (standard ring when receiving a call)
badaddr.wav ("this is a bad address")
error.wav (sound of machinery breaking)
numpad.wav (button press sound for 0,1,2,...,9 and .)
hold.wav ("holding, please stand by")
priority.wav (standard priority ring sound)
campack.wav (special ring when party is available to call)

Voice Mail

The user may record and send voice mail to remote users when the remote user's answering machine answers or calls are not completed because of the remote user being offline or busy.

Remote user is offline

When a user gets an OFFLINE from a remote *webPhone*, the user may record a voice mail message which will be e-mailed {VMAIL} to the remote *webPhone*. The voice mail file name, in order to be unique, is defined by the remote *webPhone* upon receipt of the {VMAIL}. Refer to the E-mail Communications Protocol above for details.

Upon receipt of {VMAIL}, the *webPhone* will extract the compressed audio portion of the voice mail message and save it to the webphone\vmail\in directory with the following name:

XXXXXXXX.wpm where X = {0,1,2,...9 }

The filename will be created from the *vmailName* field in *webphone.cfg.*

This nomenclature allows for 100 million unique file names before the sequence repeats itself.

Once received, the *webPhone* will update the *messages.dir* file in the ..\webphone\vmail directory. Refer to the messages.dir database schema in figures 33 - 36 for more details.

Remote user is busy

When a user gets a BUSY from a remote *webPhone*, the user may record and transmit a voice mail message to the remote *webPhone*. This transmission takes the form of multiple <Vmail> packets and a terminating <VmailEnd> packet. During the receipt of the voice mail, the remote *webPhone* is saving the voice mail message to a voice mail file named *XXXXXXXX.wpm* in the remote user's webphone\vmail\in directory.

Remote user's answering machine answers

When a user gets an ANSWERINGMACHINE from a remote *webPhone*, the remote *webPhone*'s answering machine answered the call and is playing an outgoing message to the user. Once the remote user's OGM is complete, the user may record and transmit a voice mail message to the remote *webPhone* as described above for the remote user busy condition.

Recording voice mail

When it is OK to record a voice mail message, the user's *webPhone* will activate the audio playback and record controls in the flip door of the phone. If the flip door is closed, it will be automatically opened. Once activated, the user operates the controls like a normal audio tape deck to record and playback the voice mail message. When the user is ready to transmit the voice mail message, he/she presses [END] to end the call. If the user wishes not to send a voice mail message, he/she presses [END] without having recorded a voice mail message. If the user has begun to record a voice mail message and decides he/she does not wish to send it, the user would press the cancel button [x] in the audio controls to abort the voice mail recording then press [END] to end the call.

What the user sees when voice mail arrives

The *webPhone* will increment the number of new messages in the display. If the Voice Mail Messages dialog is up, the new message will be place at the top of the list.

Copy Protection

If a user has the ITEL phone card

the *webPhone* will detect and use the card without using the webphone.reg file as a copy protection mechanism.

If the user does not have the ITEL phone card

when the user registers (i.e. pays $49.95), we will e-mail the *webphone.reg* file to the user as the special e-mail message REGISTRATION. The *webphone.reg* file contains that user's DES encrypted e-mail address. The *webPhone* will receive the REGISTRATION message and place the attached *webphone.reg* file in the webphone directory. When the registered user starts their *webPhone*, it will read the *webphone.reg* file and decrypt the user's e-mail address (This means the key is hardcoded into the *webPhone*). It will then compare the decrypted e-mail address with that in the user's *webphone.cfg* file. If the two e-mail addresses match, the *webPhone* will operate uncrippled, otherwise, it will notify the user of the problem, suggest the solution and exit.

In the event the user changes his/her e-mail address or IP address, via User Info ala [CFG] they will be required to a change of address to us (by calling ITEL on their *webPhone*, going to the ITEL Home Page or by e-mail to info@itel.com) in order to obtain a new *webphone.reg* file. A registered user with more than 2 change of address applications is suspect of copying the software.

Note: it makes no sense for a registered user to copy the software and give the it to another user since the recipient will not be able to receive phone calls or voice mail at their actual e-mail address. If the recipient changes the registered user's e-mail address and optional IP address, the *webPhone* will operate in the crippled state since the e-mail address encrypted in *webphone.reg* will not match that in *webphone.cfg*.

Configuration [CFG]

The Configuration dialog, obtained when the user pressed the [CFG] button, has the following 7 tabbed sections covering areas in which parameters are defined by the user to control the operation of the *webPhone*. Refer to figures i - i.

1. User Information
2. Phone
3. Answering Machine
4. Phone Directory
5. Sound Effects
6. Audio Card
7. Activity Log

All the configuration information is stored in the *webphone.cfg* file located in the webphone directory.

ITEL Home Page

The ITEL Home Page consists of

- a brief description of the Internet Telephone Company
- a succinct description of our product's features and how it is vastly superior to Vocaltec's iphone and is less expensive.
- a graphical (e.g. image of *webphone*) and textual link to a detailed description of the *webPhone's* features
- a graphical link and textual link to FTP the crippled *webPhone* to the user
- a graphical and textual link to the ***order*** form
- a graphical and textual link to the ***change of address*** form
- a graphical and textual link to ***directory assistance*** form
- a link to WEBPALS description, registration and inquiry form

Information (Directory Assistance)
Enables users to query the master phone directory for other user's e-mail and IP addresses (if known). This will initially be a free service.

Change of Address
Enables the user to enter their old e-mail address and IP address (if known) then prompts the user to enter their new e-mail address and IP address (if known). If the user has already had less than two prior change of address requests, ITEL will email the user his/her new *webphone.reg* file. If the user has already had two change of address requests, ITEL will email the user an explanation request form which must be completed and emailed back to ITEL. If the explanation is valid, ITEL will email the user his/her new *webphone.reg* file. If the explanation is suspect, ITEL will inform the user it is against the law to copy licensed software and he/she will need to purchase another *webPhone*.

WebPhone Protocol (WPP) Packet Definitions

Packet #	Packet	Packet Type	Direction	Data
100	Invalid	WPP_INVALID	← →	WPP_INVALID
101	Online Req	WPP_ONLINEREQ	→	WPP_ONLINEREQ, sid, version, emailAddr, IPAddr, onlineState
102	OnlineACK	WPP_ONLINEACK	←	WPP_ONLINEACK, sid, onlineStatus
103	Offline	WPP_OFFLINE	← →	WPP_OFFLINE, sid
104	Hello	WPP_HELLO	← →	WPP_HELLO, sid, version
105	Connect Req	WPP_CONNECTREQ	→	WPP_CONNECTREQ, sid, version, callType, partyEmailAddr, emailAddr, IPAddr, connectState
106	Connect ACK	WPP_CONNECTACK	← →	WPP_CONNECTACK, sid,connectStatus, partyIPaddr
107	Call	WPP_CALL	← →	WPP_CALL, sid, version, emailAddr, IPAddr, userInfo
108	CallACK	WPP_CALLACK	← →	WPP_CALLACK, sid, version, emailAddr, IPAddr, userInfo
109	CnfCall	WPP_CNFCALL	← →	WPP_CNFCALL, sid, version, emailAddr, IPAddr, userInfo
110	CnfCallACK	WPP_CNFCALLACK	← →	WPP_CNFCALLACK, sid, version
111	Answer	WPP_ANSWER	← →	WPP_ANSWER, sid
112	Busy	WPP_BUSY	← →	WPP_BUSY, sid
113	AnsMachine	WPP_ANSMACH	← →	WPP_ANSMACH, sid, state
114	End	WPP_END	← →	WPP_END, sid
115	Hold	WPP_HOLD	← →	WPP_HOLD, sid, (ON | OFF)
116	Reject	WPP_REJECT	← →	WPP_REJECT, sid
117	Camp	WPP_CAMP	← →	WPP_CAMP, sid
118	CampACK	WPP_CAMPACK	← →	WPP_CAMPACK, sid
119	Audio	WPP_AUDIO	← →	WPP_AUDIO, sid, audioType, silence, length, audioData
120	Vmail	WPP_VMAIL	← →	WPP_AUDIO, sid, audioType, silence, length, audioData
121	VmailEnd	WPP_VMAILEND	← →	WPP_VMAILEND, sid
122	OgmEnd	WPP_OGMEND	← →	WPP_OGMEND, sid
123	CnfAdd	WPP_CNFADD	← →	WPP_CNFADD, sid, partyEmailAddr, partyIPaddr, partInfo
124	CnfDrop	WPP_CNFDROP	← →	WPP_CNFDROP, sid
125	FileXmtReq	WPP_FILEXMTREQ	← →	WPP_FILEXMTREQ, sid, fileType, fileName, fileSize

WebPhone Protocol (WPP) Packet Definitions (con't)

Packet #	Packet	Packet Type	Direction	Data
126	FileXmtAck	WPP_FILEXMTACK	← →	WPP_FILEXMTACK, sid
127	File	WPP_FILE	← →	WPP_FILE, sid, length, fileData
128	FileXmtEnd	WPP_FILEXMTEND	← →	WPP_FILEXMTEND, sid
129	FileXmtAbort	WPP_FILEXMTABORT	← →	WPP_FILEXMTABORT, sid
130	InfoReq	WPP_INFOREQ	→	WPP_INFOREQ, sid, query
131	InfoACK	WPP_INFOACK	←	WPP_INFOACK, sid, nparties
132	Info	WPP_INFO	←	WPP_INFO, sid, partyInfo
133	InfoAbort	WPP_INFOABORT	→	WPP_INFOABORT, sid
134	UserInfoReq	WPP_USRINFOREQ	←	WPP_USRINFOREQ, sid
135	UserInfo	WPP_USRINFO	→	WPP_USRINFO, sid, version, userInfo
136	WBImageStart	WPP_WBIMAGESTART	←	WPP_WBIMAGESTART, sid, fileSize, imageType
137	WBImage	WPP_WBIMAGE	←	WPP_WBIMAGE, sid, length, imageData
138	WBImageEnd	WPP_WBIMAGEEND	←	WPP_WBIMAGEEND, sid
139	WBAudioStart	WPP_WBAUDIOSTART	←	WPP_WBAUDIOSTART, sid, fileSize, audioType
140	WBAudio	WPP_WBAUDIO	←	WPP_WBAUDIO, sid, length, audioData
141	WBAudioEnd	WPP_WBAUDIOEND	←	WPP_WBAUDIOEND, sid
142	Registration	WPP_REG	←	WPP_REG, sid, EEmailAddr
143	Caller OK	WPP_CALLEROK	←	WPP_CALLEROK, sid, version, emailAddr
144	Caller ACK	WPP_CALLERACK	←	WPP_CALLERACK,sid, callerStatus
145	Key Pad	WPP_KEYPAD	←	WPP_KEYPAD,sid,(ON | OFF)
146	Key	WPP_KEY	→	WPP_KEY,sid,ascii character

WebPhone Protocol (WPP) Packet Data Definitions

Element	Data Type	Comment
WPP_*	unsigned char	WPP_ message identifer
sid	unsigned long	session id unique per call
version	unsigned short (3)	version of the webphone (capability, protocol, vendor)
emailAddr	varchar(90)	email address of caller
IPAddr	varchar(80)	IP Address
onlineState	unsigned char	bit 0 (ACTIVE | INACTIVE)
		bit 1 (Merchant Phone)
		bit 2 (Connection Server)
		bit 3 (webboard disabled)
		bit 4 Not Used
		bit 5 Not Used
		bit 6 Not Used
		bit 7 Not Used
callType	unsigned char	call type 0: EMAIL | 1: IPCALL
partyEmailAddr	varchar(90)	email address of person to call
connectStatus	unsigned char	0: NOWEBPHONE
		1: ONLINE
		2: OFFLINE
		3: RECONNECT
		4: PERM_RECONNECT
partyIPAddr	varchar(80)	IP Address of person to call
userInfo	varchar(120)	firstName, LastName, alias, emailAddr, street, apt, city, state, country, postalCode, phone, fax, company
audioType	usigned char	audio compress type
		0:GSM
		1: TRUESPEECH

WebPhone Protocol (WPP) Packet Data Definitions (con't)

Element	Data Type	Comment
length	unsigned short	length of audio or data in bytes
audioData	512 Bytes	compressed audio data
fileType	unsigned char	file type
		0:DATA
		1:EMAIL
		2:TEXT
		3:BINARY
fileName	varchar(13)	name of file to be transmitted. 8.3 nomenclature
fileSize	unsigned long	size of file to be transmitted in bytes
fileData	variable	file data
query	varchar(120)	firstName, lastName, company, city, state, country
nparties	unsigned long	number of parties or query records being sent
size	unsigned long	size of file (IMAGE or AUDIO) to be sent
imageType	unsigned char	image type
		0: GIF
		1: JPG
imageData	512 Bytes	image data
eemailAddr	varchar(90)	encrypted email Address
onlineStatus	unsigned char	0 OK
		-1 Error
callerStatus	unsigned char	0 is unpaid
		1 if paid
onlineState	unsigned char	bit 0 webboard disabled
		bit 1 Not Used
		bit 2 Not Used
		bit 3 Not Used
		bit 4 Not Used
		bit 5 Not Used
		bit 6 Not Used
		bit 7 Not Used

Customer Table

Field	Data Type	Ctree Type	Index	Offset	Comments
delflag	int	COUNT		0	Used by Database
id	ulong	LONG	Y	2	Unique ID Sequence
activated	char	char	Y	6	0 = NO, 1 = YES
activationDate	ulong	LONG		7	Secs since 00:00 GMT Jan 1, 1970
version capability	ushort	COUNT		11	Version of the Webphone
version protocol	ushort	COUNT		13	
version vendor	ushort	COUNT		15	
paid	char	char		17	0 = NO, 1 = YES
prePaidCode	varchar(16)	TEXT[16]	Y	18	
firstName	varchar(10)	TEXT[10]	Y	34	
lastName	varchar(25)	TEXT[25]	Y	44	
alias	varchar(20)	TEXT[20]		69	
emailAddr	varchar(90)	TEXT[90]	Y	89	
IPAddr	varchar(80)	TEXT[80]		179	0.0.0.0 if not known
street	varchar(50)	TEXT[50]		259	
apt	varchar(5)	TEXT[5]		309	
city	varchar(20)	TEXT[20]	Y	314	
state	varchar(20)	TEXT[20]	Y	334	
country	varchar(20)	TEXT[20]	Y	354	
postalCode	varchar(20)	TEXT[20]		374	
phone	varchar(25)	TEXT[25]		394	
fax	varchar(25)	TEXT[25]		419	
company	varchar(25)	TEXT[25]	Y	444	Company Name
addrChanges	char	char		469	Number of address changes
addrChangeDate	ulong	LONG		470	Secs since 00:00 GMT Jan 1, 1970
publish	char	char		474	0 = NO, 1 = YES
accessDate	ulong	LONG		475	Secs since 00:00 GMT Jan 1, 1970
accessCount	ulong	LONG		479	# of times user has started Webphone
callCount	ulong	LONG		483	Total number of outbound calls customer has made

Total Record Size = 487

Online Table

Field	Data Type	Ctree Type	Index	Offset	Comments
delflag	int	COUNT		0	Used by Database
emailAddr	varchar(90)	TEXT[90]	Y	2	
IPAddr	varchar(80)	TEXT[80]	Y	92	
flags	char	char		172	
onlineDate	ulong	LONG		174	

Total Record Size = 178

WebBoard Table

Field	Data Type	Ctree Type	Index	Offset	Comments
delflag	int	COUNT		0	Used by Database
id	ulong	LONG	Y	2	Unique ID Sequence
image	varchar(8)	TEXT[8]		6	Filename of image file
imageType	char	char		14	.GIF =0, JPG = 1
audio	varchar(8)	TEXT[8]		15	Filename of TSP encoded .WAV file
audioType	char	char		23	GSM = 0, TRUESPEECH = 1
hits	ulong	LONG		24	Number of accrued hits

Total Record Size = 28

Weboard Config Table

Field	Data Type	Ctree Type	Index	Offset	Comments
delflag	int	COUNT		0	Used by Database
count	ulong	LONG	Y	2	Number of WebBoards

Total Record Size = 6

Advertiser Table

Field	Data Type	Ctree Type	Index	Offset	Comments
delflag	int	COUNT		0	Used by ctree
id	ulong	LONG	Y	2	Unique ID
weboardID	ulong	LONG		6	Link to WebBoard record
name	varchar(50)	TEXT[50]		10	Company's name
url	varchar(80)	TEXT[80]		60	URL to Home Page
street	varchar(50)	TEXT[50]		140	
apt	varchar(5)	TEXT[5]		190	
city	varchar(20)	TEXT[20]		195	
state	varchar(20)	TEXT[20]		215	
country	varchar(20)	TEXT[20]		235	
postalCode	varchar(20)	TEXT[20]		255	
phone	varchar(25)	TEXT[25]		275	
fax	varchar(25)	TEXT[25]		300	
contact	varchar(35)	TEXT[35]		325	Name of contact

Total Record Size = 360

Point to Point calling Mechanism

The diagram in figure 38 illustrates the mechanism by which the *webPhone* places calls and connects to other *webPhone* users who are connected to the internet via dialup SLIP/PPP lines via their 14.4/28.8 modems.

If the remote *webPhone* has a fixed IP address, the user transmits *<Call>* thereby bypassing the *ConnectRequest/ConnectOK* steps to establish a connection. *WebPhones* always maintain 1 open socket listening for a *Call*. Therefore, if all 4 lines are in use, the *webPhone* will send back a *<Busy>* to the caller.

Calling Scenarios

1. Recipient is offline
   initiator times out, kills socket, plays offline.wav
   initiator can e-mail *{VMAIL}*

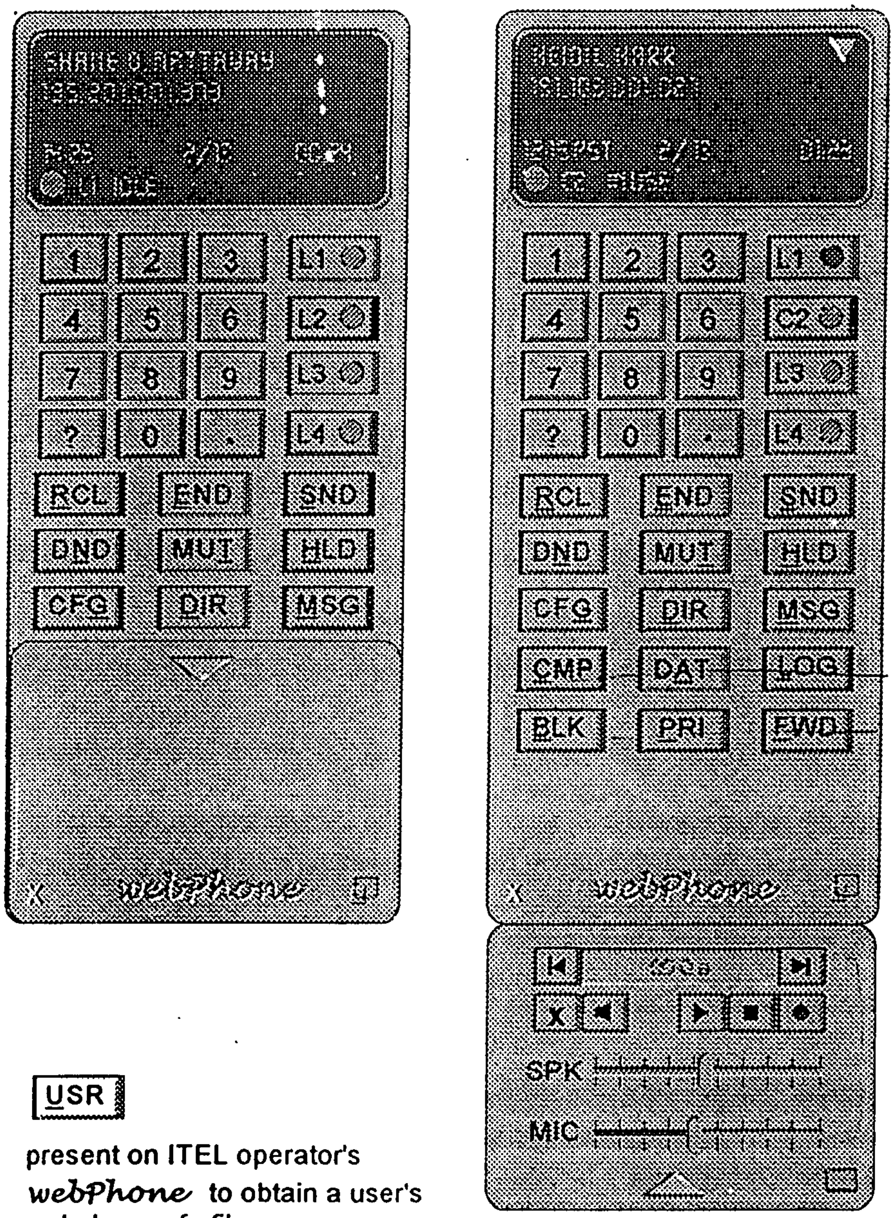
2. Recipient has all 4 lines in use
   recipient sends back *Busy*, initiator plays busy.wav
   initiator can transmit *<Vmail>*

3. Recipient is on-line but does not answer
   initiator times out on *<ConnectOK>*, recipient's answering machine plays ogm.wav
   initiator can transmit *<Vmail>*

4. Recipient goes offline after transmitting *<ConnectOK>*
   initiator fails on transmitting *<Call>*, plays offline.wav
   initiator can e-mail *{VMAIL}*

5. Initiator goes offline after sending *{CALL}* and another *webPhone* gets the same IP address assigned and receives the *<ConnectOK>* from the recipient (extremely low probability of occurrence)
   only if the new initiator has an open socket listening for a *<ConnectOK>* from another party will he/she receive the *<ConnectOK>* from the wrong party, the initiator checks the session number in the *<ConnectOK>* and discovers the mismatch and disregards the transmission.
   in any event, the recipient will time out on *<Call>*

6. Recipient or initiator goes offline during the conversation
   failure on read/write to socket occurs both parties announce offline and can e-mail *{VMAIL}*.



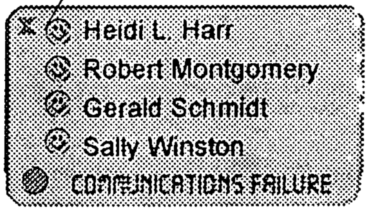
USR

present on ITEL operator's *webPhone* to obtain a user's webphone.cfg file

Figure 1

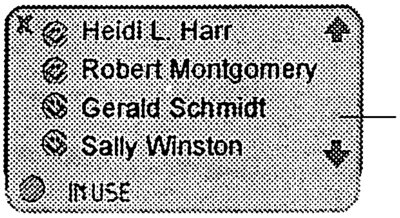
different icon for remote hold

CNF list
PRI list
BLK list
CMP list



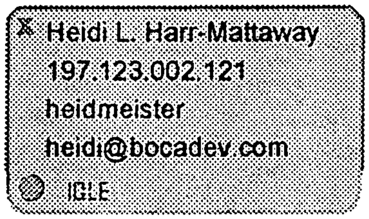
Heidi L. Harr
Robert Montgomery
Gerald Schmidt
Sally Winston
COMMUNICATIONS FAILURE

CNF list
PRI list
BLK list
CMP list



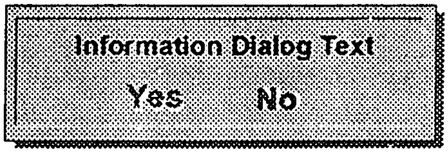
Heidi L. Harr
Robert Montgomery
Gerald Schmidt
Sally Winston
IN USE

more than 5 in list

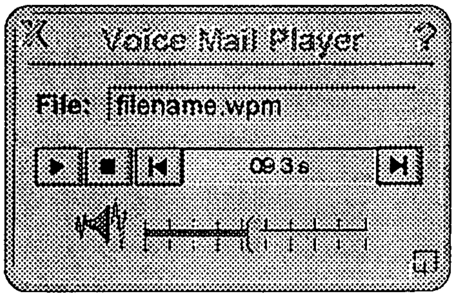


Heidi L. Harr-Mattaway
197.123.002.121
heidmeister
heidi@bocadev.com
IDLE

speed dial party info
Line party info
FWD party info
ALL party info



Information Dialog Text



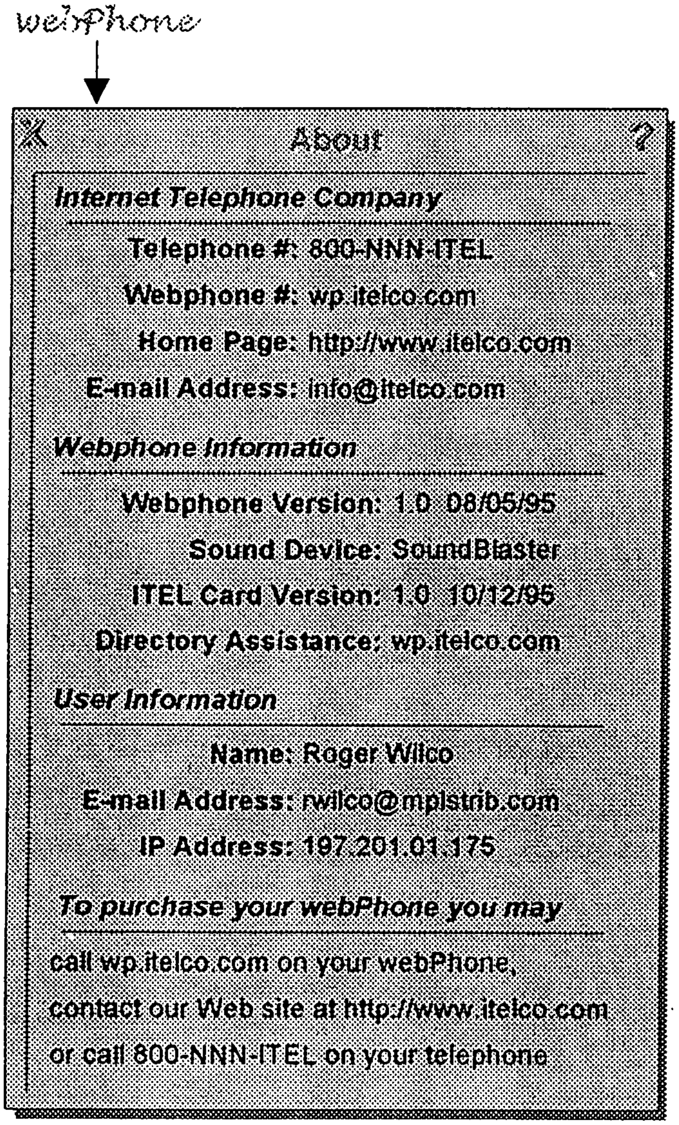
Voice Mail Player

File: filename.wpm

093s

wpmplay.exe

plays saved voice mail files
located in the file system



Internet Telephone Company

Telephone #: 800-NNN-ITEL
Webphone #: wp.itelco.com
Home Page: http://www.itelco.com
E-mail Address: info@itelco.com

Webphone Information

Webphone Version: 1.0 08/05/95
Sound Device: SoundBlaster
ITEL Card Version: 1.0 10/12/95
Directory Assistance: wp.itelco.com

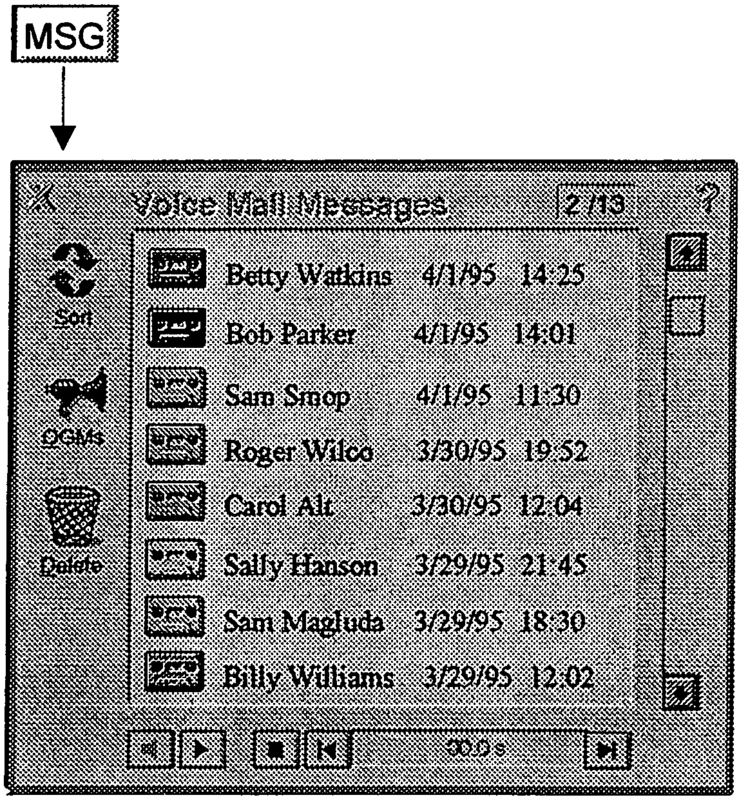
User Information

Name: Roger Wilco
E-mail Address: rwilco@mplstrib.com
IP Address: 197.201.01.175

To purchase your webPhone you may

call wp.itelco.com on your webPhone,
contact our Web site at http://www.itelco.com
or call 800-NNN-ITEL on your telephone

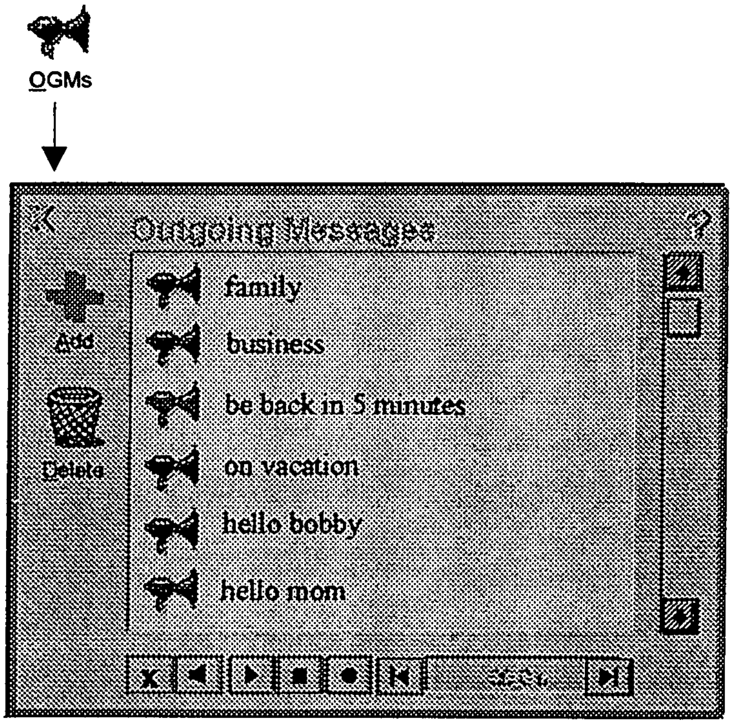
Figure 3



Voice Mail Messages 2/13

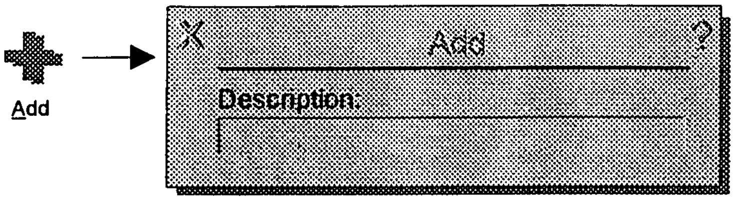
Betty Watkins	4/1/95	14:25
Bob Parker	4/1/95	14:01
Sam Smop	4/1/95	11:30
Roger Wilco	3/30/95	19:52
Carol Alt	3/30/95	12:04
Sally Hanson	3/29/95	21:45
Sam Magluda	3/29/95	18:30
Billy Williams	3/29/95	12:02

***dbl click** - playback all selected*
***left click (ctrl left click)** - select/deselect*
***Alt-left click** - select/deselect All*
***right click** - message details*
***drag** to move to File Manager dir*
or append to another vmail msg
***Ctrl-drag** to copy to FIle Manager dir*



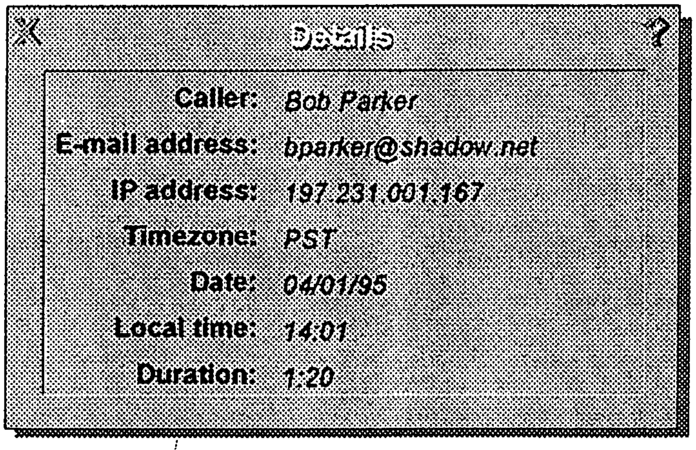
dbl click *- playback all selected*
left click *- select/deselect*
Alt-left click *- select/deselect All*
right click *- OGM details*
drag *to DIR entry to assign OGM*

Figure 5



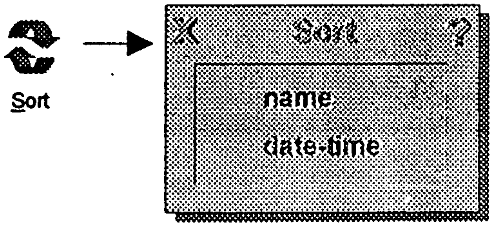
Add

Description:



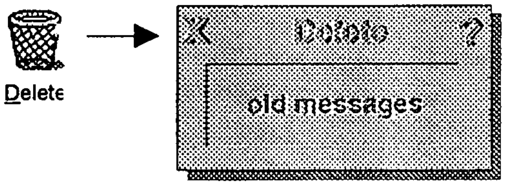
Details

Caller: *Bob Parker*
E-mail address: *bparker@shadow.net*
IP address: *197.231.001.167*
Timezone: *PST*
Date: *04/01/95*
Local time: *14:01*
Duration: *1:20*



Sort

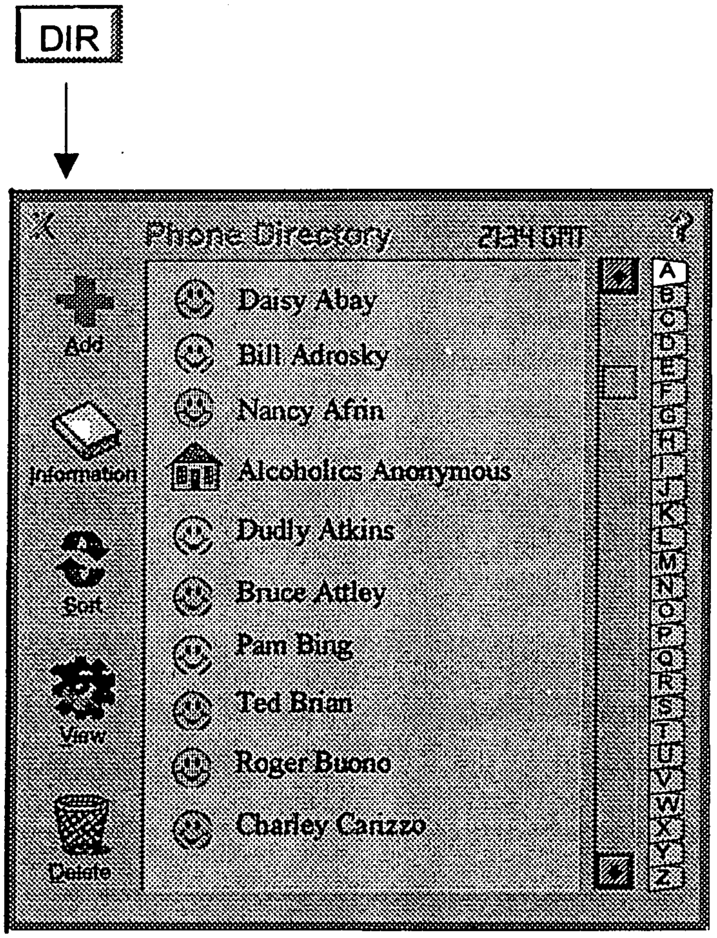
name
date-time



Delete

old messages

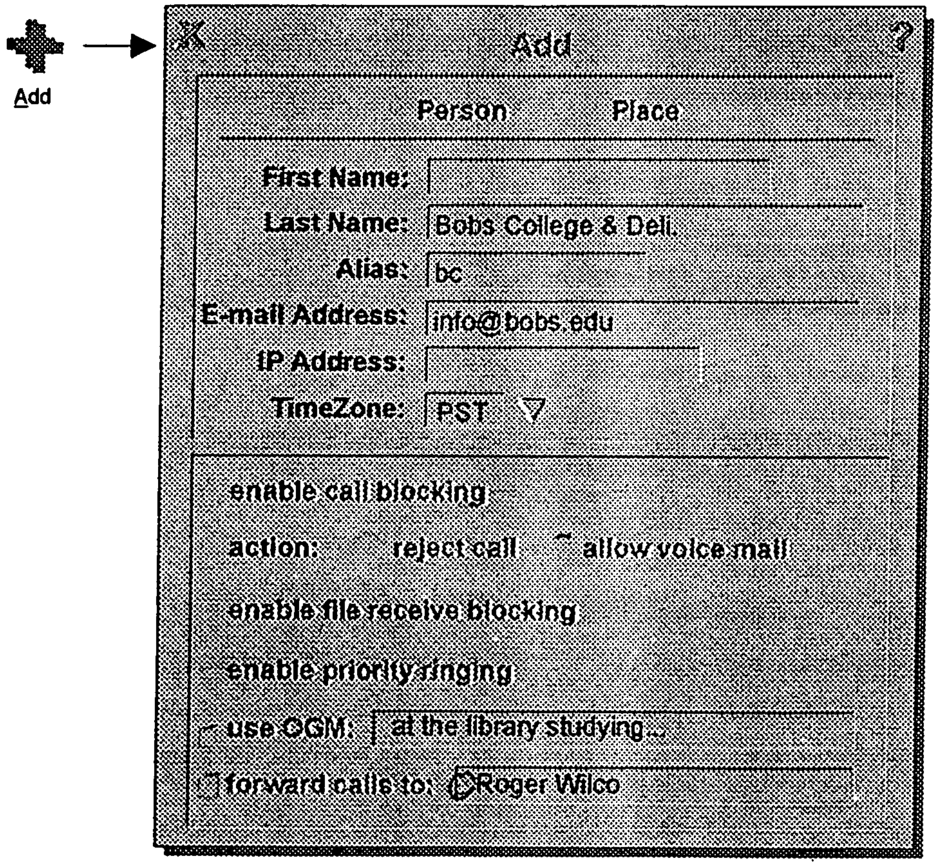
Figure 6



Daisy Abay
Bill Adrosky
Nancy Afrin
Alcoholics Anonymous
Dudly Atkins
Bruce Attley
Pam Bing
Ted Brian
Roger Buono
Charley Canzzo

☺

left click - select/deselect entry
Alt-click - select/seselect All entries
dbl click - call entry
right click -update entry
drag to number pad position for speed dial
or to [FWD] to assign to call forwarding
or to idle [Ln] to call on that line



First Name:

Last Name: Bobs College & Deli.

Alias: bc

E-mail Address: info@bobs.edu

IP Address:

TimeZone: PST ▽

enable call blocking

action: reject call allow voice mail

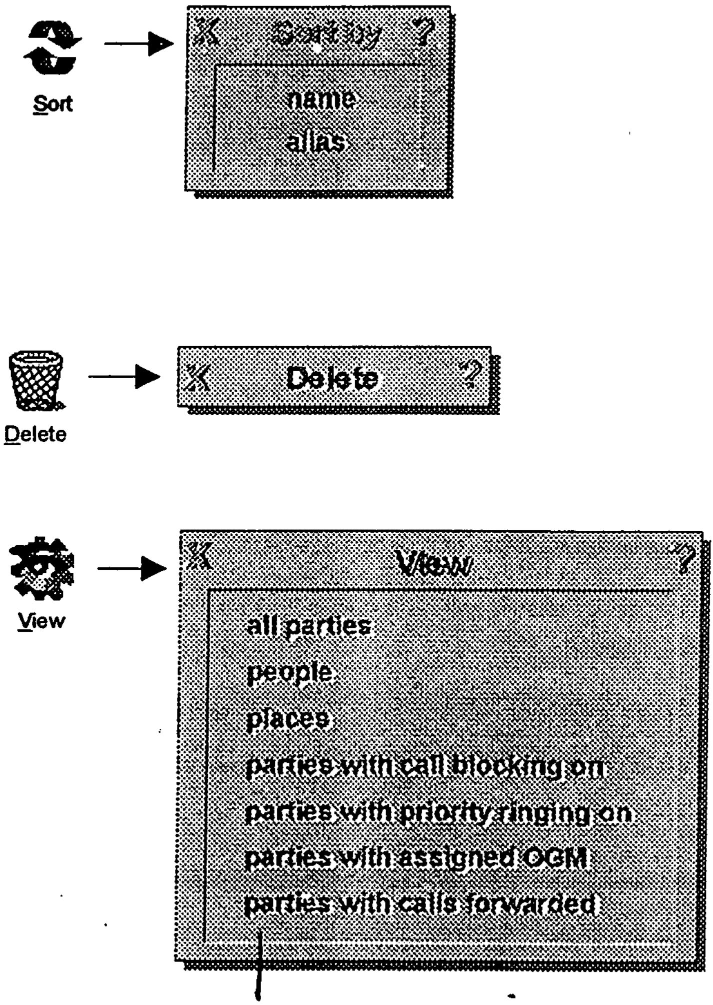
enable file receive blocking

enable priority ringing

use OGM: at the library studying...

□ forward calls to: Roger Wilco

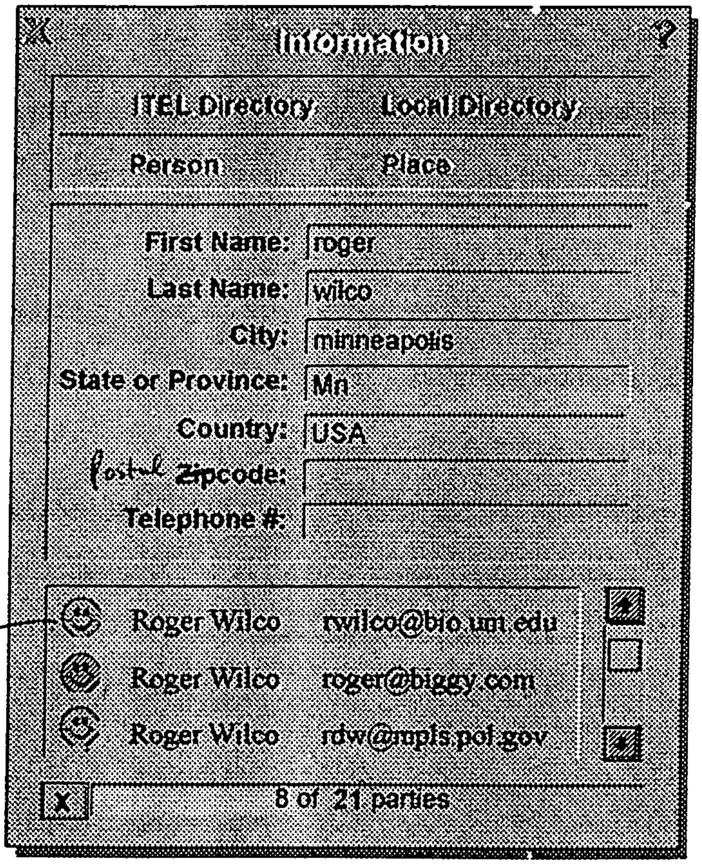
Figure 8


Sort
Sort by
name
alias
Delete
Delete
View
View
all parties
people
places
parties with call blocking on
parties with priority ringing on
parties with assigned OGM
parties with calls forwarded

parties w/ file blocking on

% any num chars follow
? any single char

Information



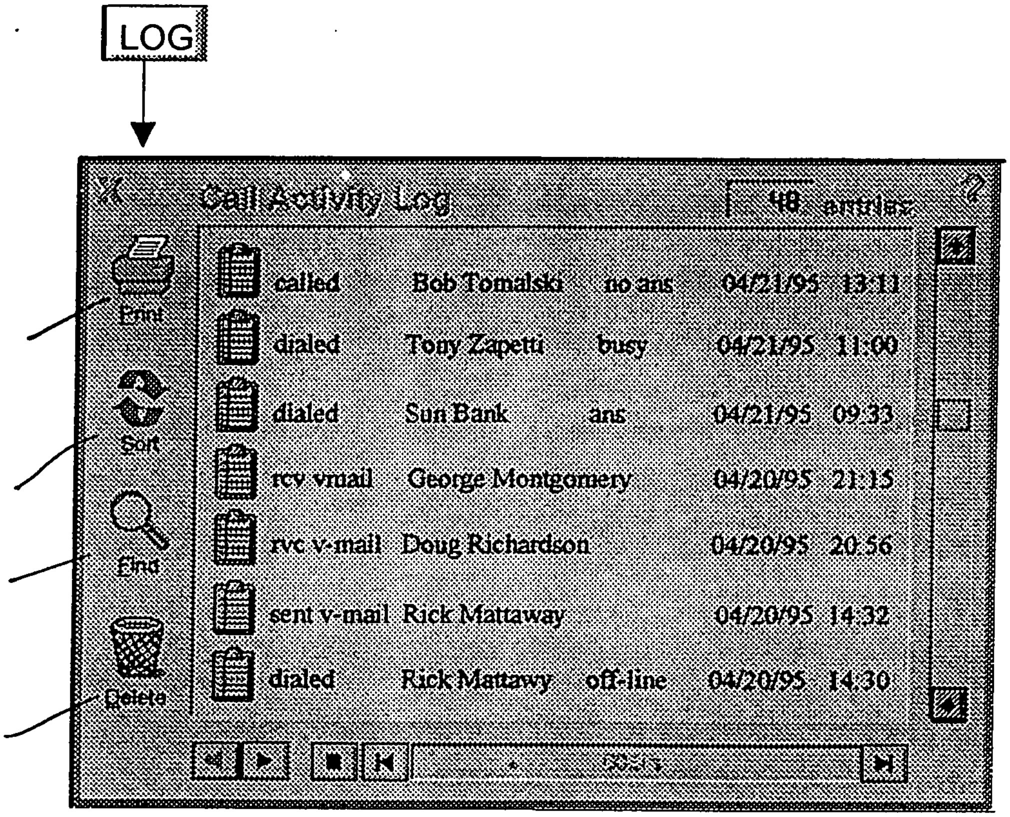
First Name:	roger
Last Name:	wilco
City:	minneapolis
State or Province:	Mn
Country:	USA
Postal ~~Zip~~code:	
Telephone #:	

☺	Roger Wilco	rwilco@bio.umn.edu
☺	Roger Wilco	roger@biggy.com
☺	Roger Wilco	rdw@mpls.pol.gov

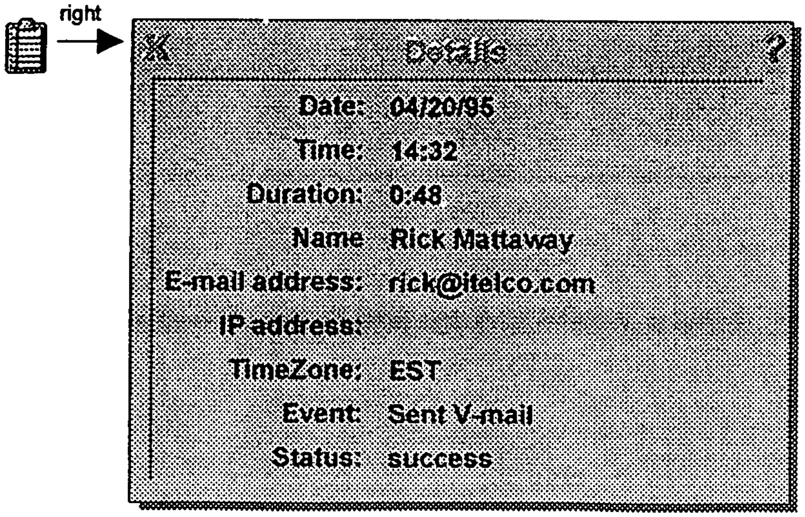
X 8 of 21 parties

Release 3.0

Dir add

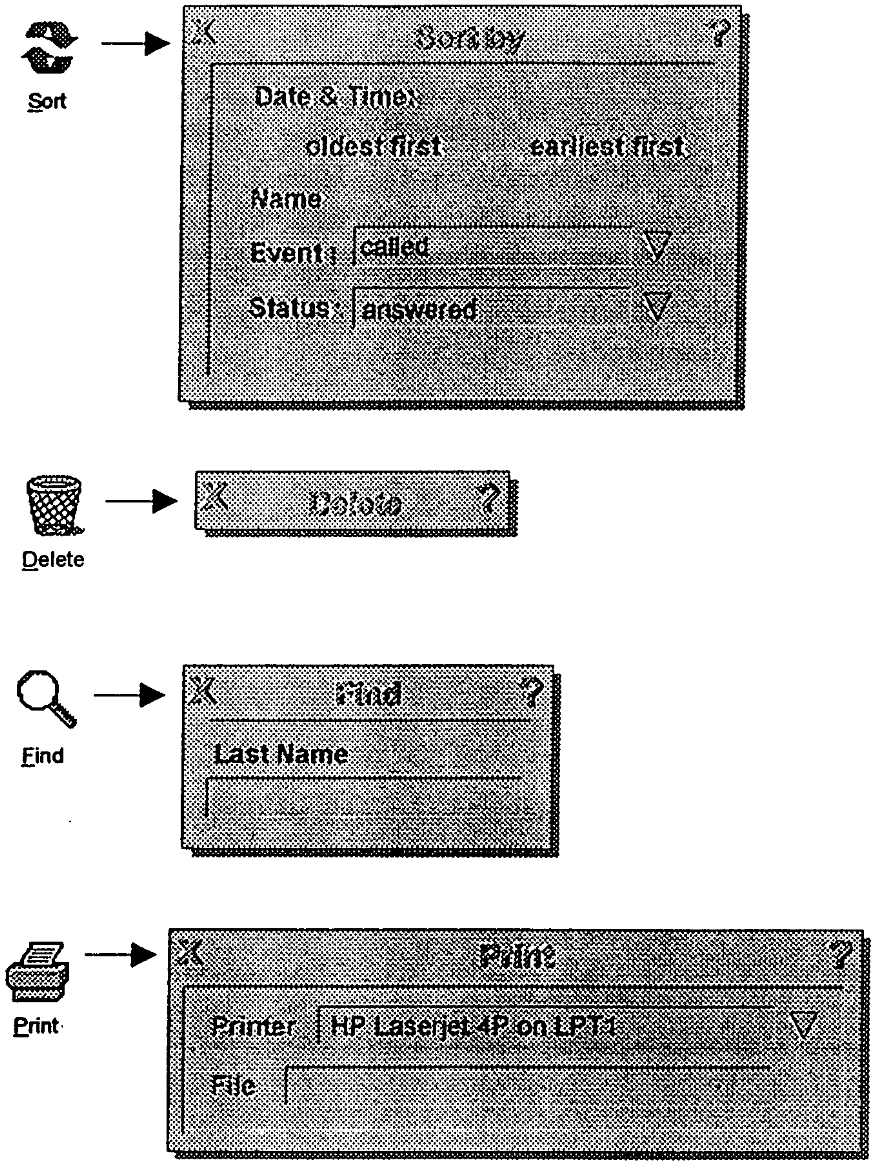


called	Bob Tomalski	no ans	04/21/95	13:11
dialed	Tony Zapetti	busy	04/21/95	11:00
dialed	Sun Bank	ans	04/21/95	09:33
rcv vmail	George Montgomery		04/20/95	21:15
rvc v-mail	Doug Richardson		04/20/95	20:56
sent v-mail	Rick Mattaway		04/20/95	14:32
dialed	Rick Mattawy	off-line	04/20/95	14:30



Date: 04/20/95
Time: 14:32
Duration: 0:48
Name Rick Mattaway
E-mail address: rick@itelco.com
IP address:
TimeZone: EST
Event: Sent V-mail
Status: success

Figure 11

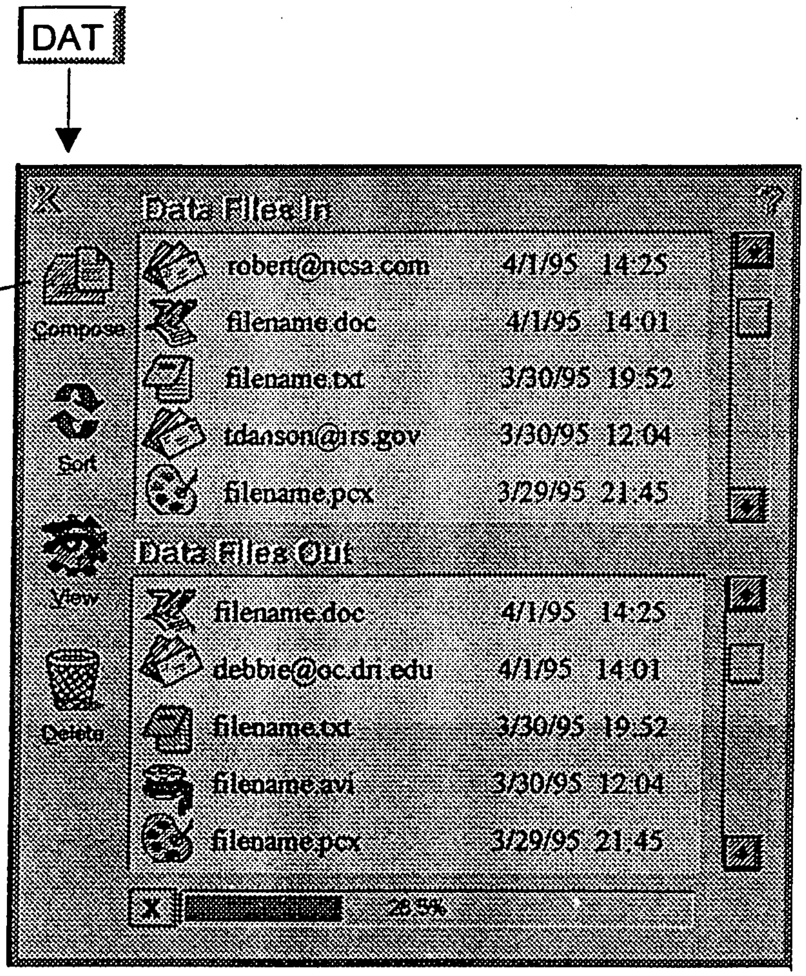


Event: called

Status: answered

Printer: HP Laserjet 4P on LPT1

Figure 12



Release 3.f

Data Files In

robert@ncsa.com	4/1/95	14:25
filename.doc	4/1/95	14:01
filename.txt	3/30/95	19:52
tdanson@irs.gov	3/30/95	12:04
filename.pcx	3/29/95	21:45

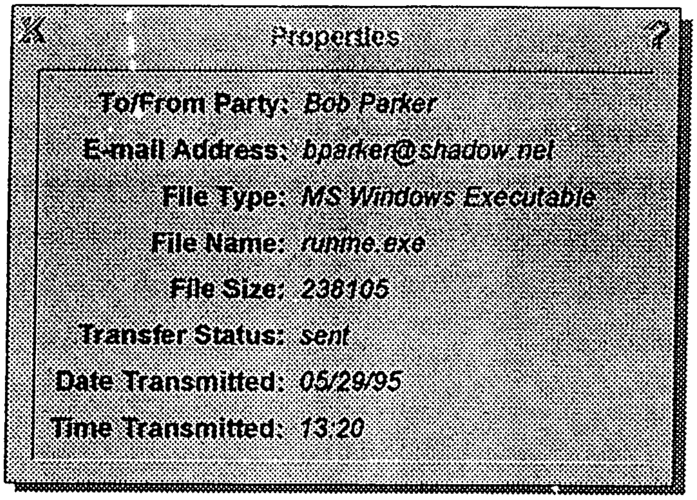
Data Files Out

filename.doc	4/1/95	14:25
debbie@oc.dri.edu	4/1/95	14:01
filename.txt	3/30/95	19:52
filename.avi	3/30/95	12:04
filename.pcx	3/29/95	21:45

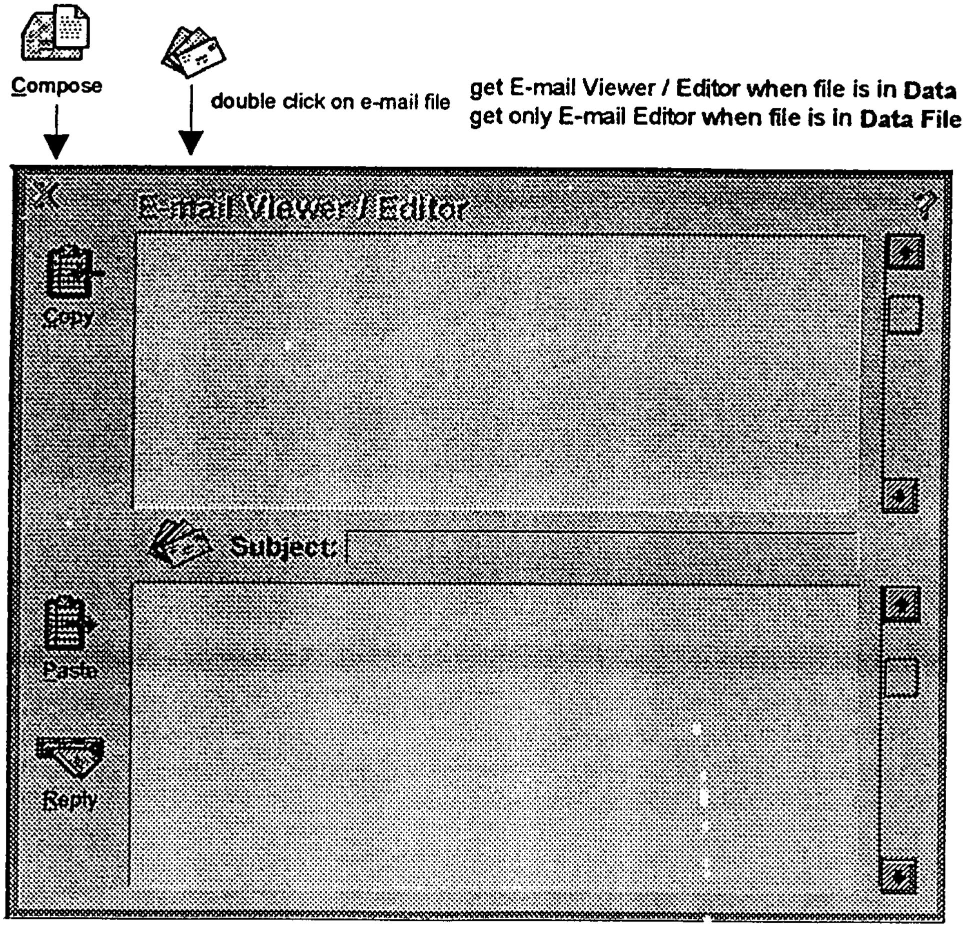
CONFIDENTIAL INFORMATION

Figure 13

right click on file to obtain properties:



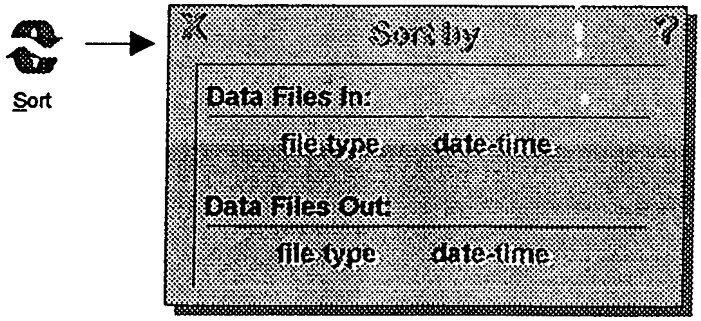
To/From Party: *Bob Parker*
E-mail Address: *bparker@shadow.net*
File Type: *MS Windows Executable*
File Name: *runme.exe*
File Size: *238105*
Transfer Status: *sent*
Date Transmitted: *05/29/95*
Time Transmitted: *13:20*



double click on e-mail file

get E-mail Viewer / Editor when file is in **Data Files In**
get only E-mail Editor when file is in **Data Files Out**

Figure 14



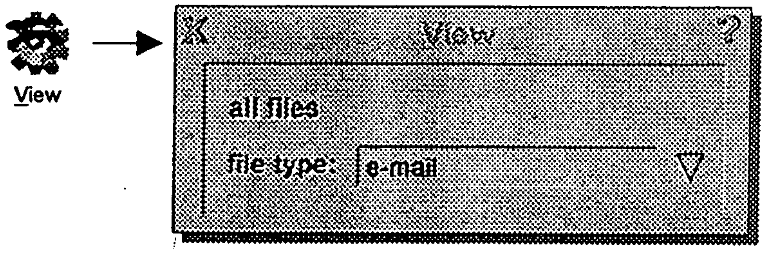
Sort by

Data Files In:

file type date-time

Data Files Out:

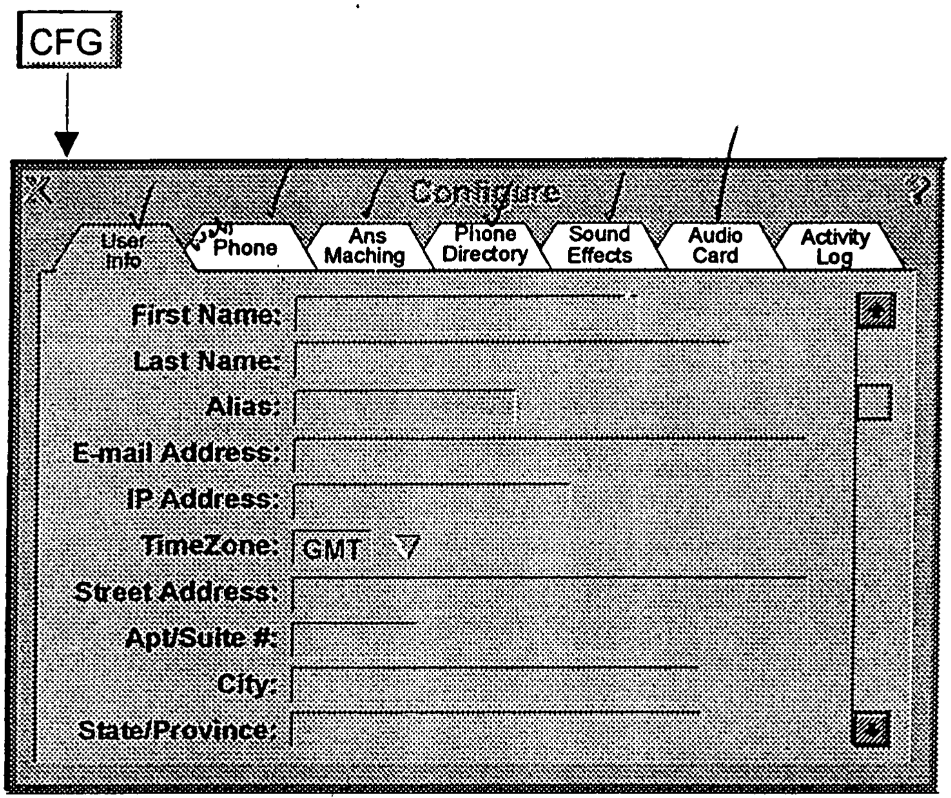
file type date-time



View

all files

file type: e-mail ▽



First Name:

Last Name:

Alias:

E-mail Address:

IP Address:

TimeZone: GMT

Street Address:

Apt/Suite #:

City:

State/Province:

Zipcode:

Country:

Telephone #:

Fax #:

Company Name:

Figure 16

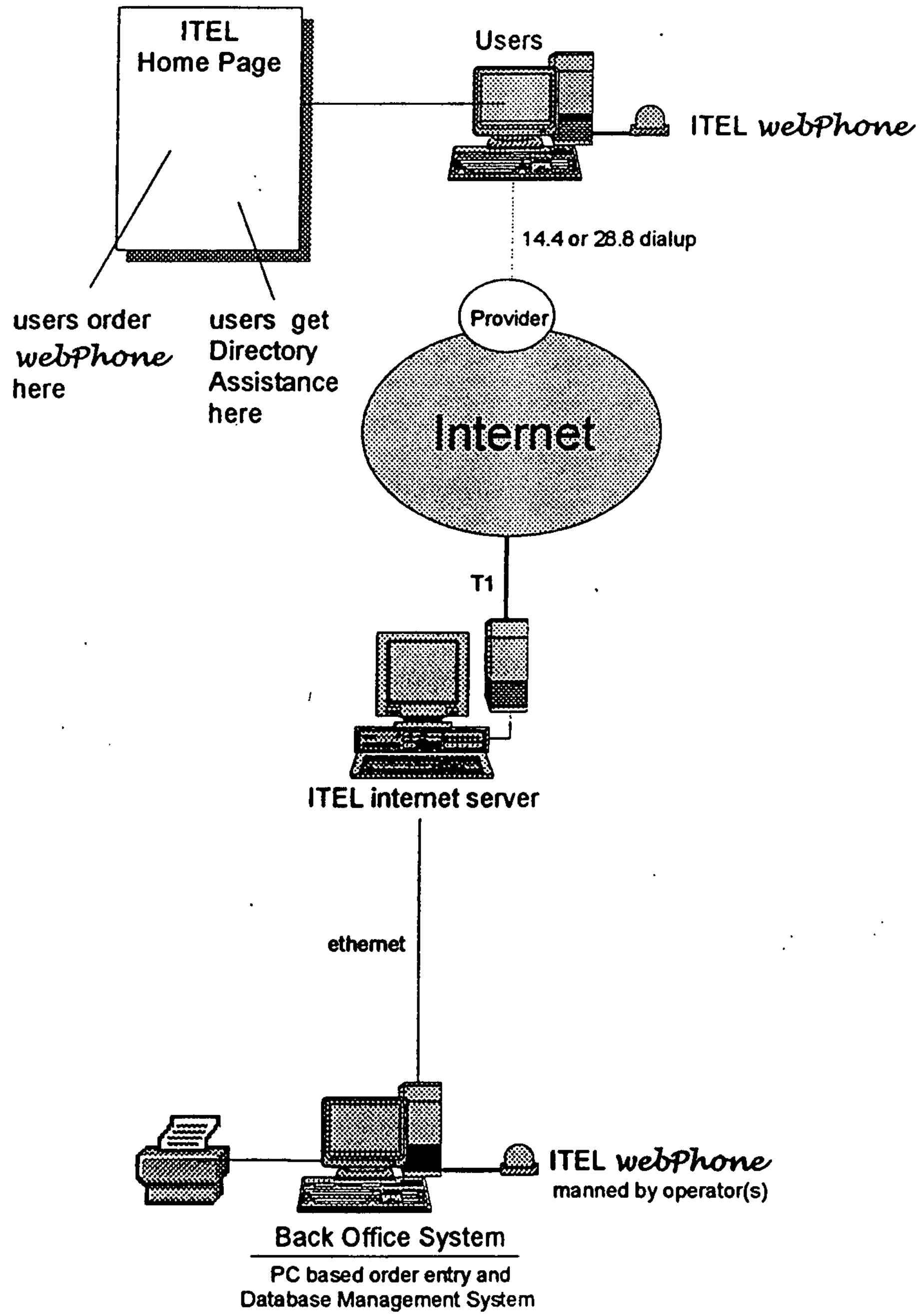

ITEL
Home Page
Users
ITEL webPhone
14.4 or 28.8 dialup
users order webPhone here
users get Directory Assistance here
Provider
Internet
T1
ITEL internet server
ethernet
ITEL webPhone
manned by operator(s)
Back Office System
PC based order entry and Database Management System

Figure 26

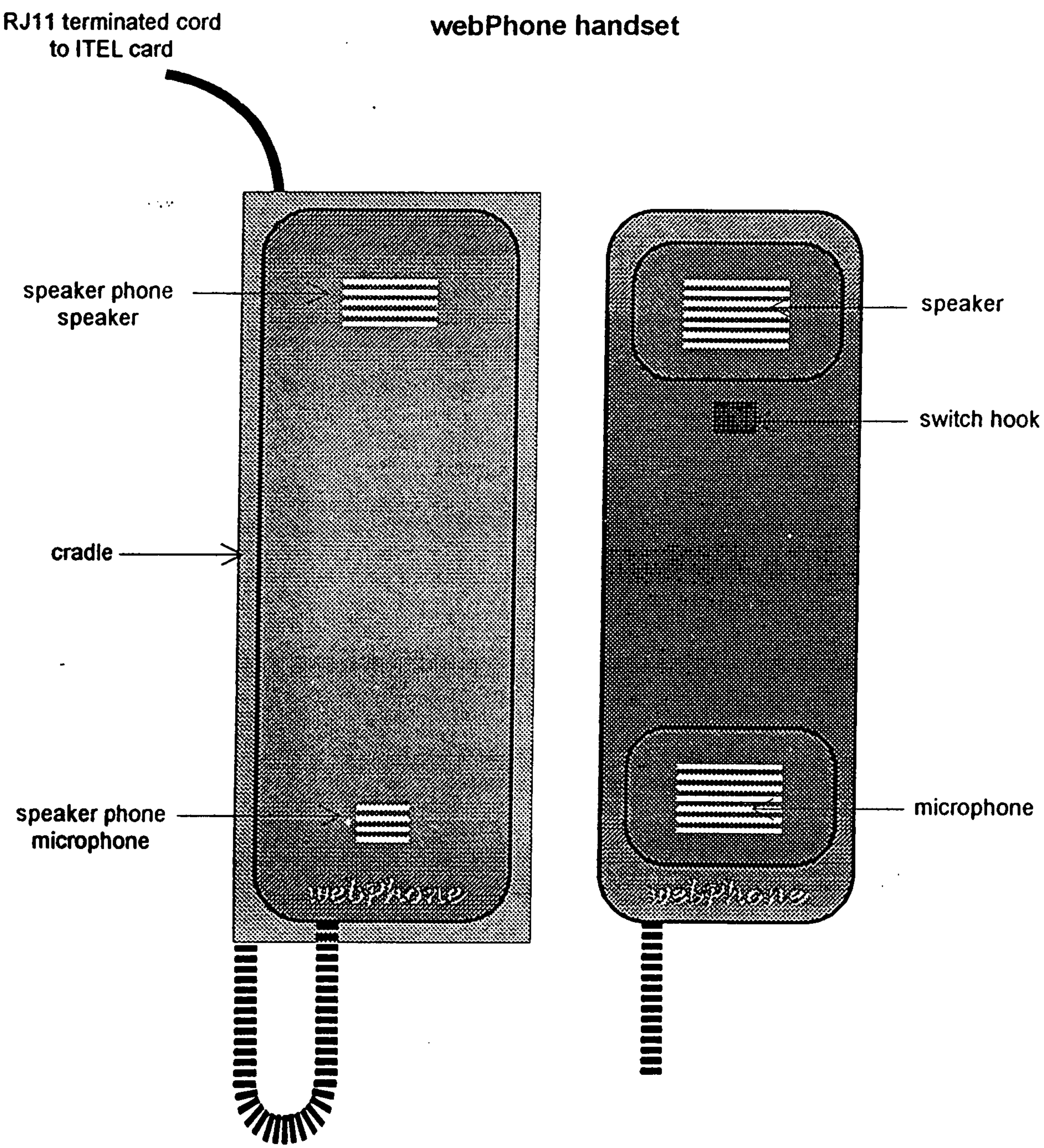

webPhone handset
RJ11 terminated cord to ITEL card
speaker phone speaker
cradle
speaker phone microphone
webPhone
speaker
switch hook
microphone
webPhone

Figure 27

System Architecture

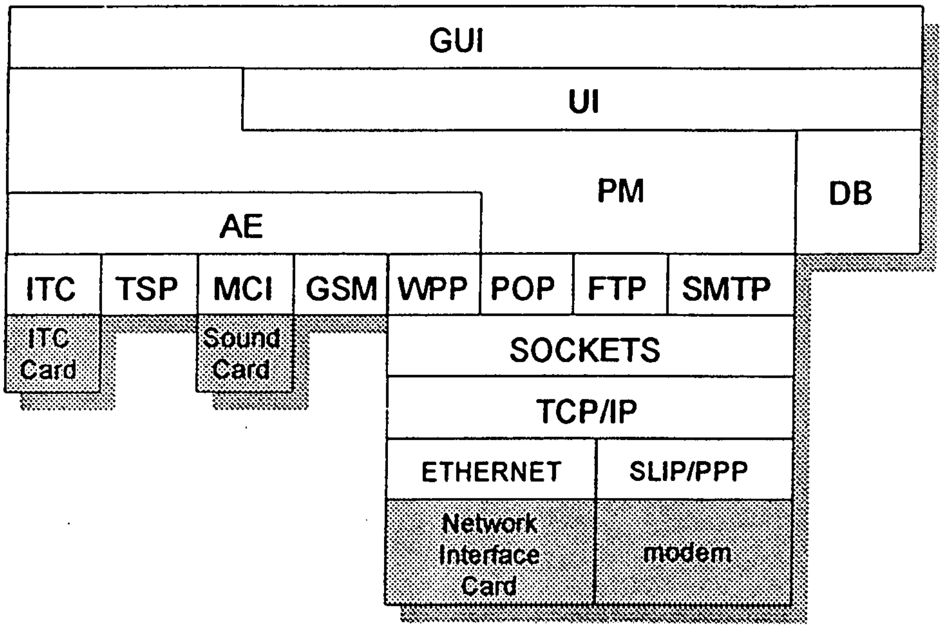
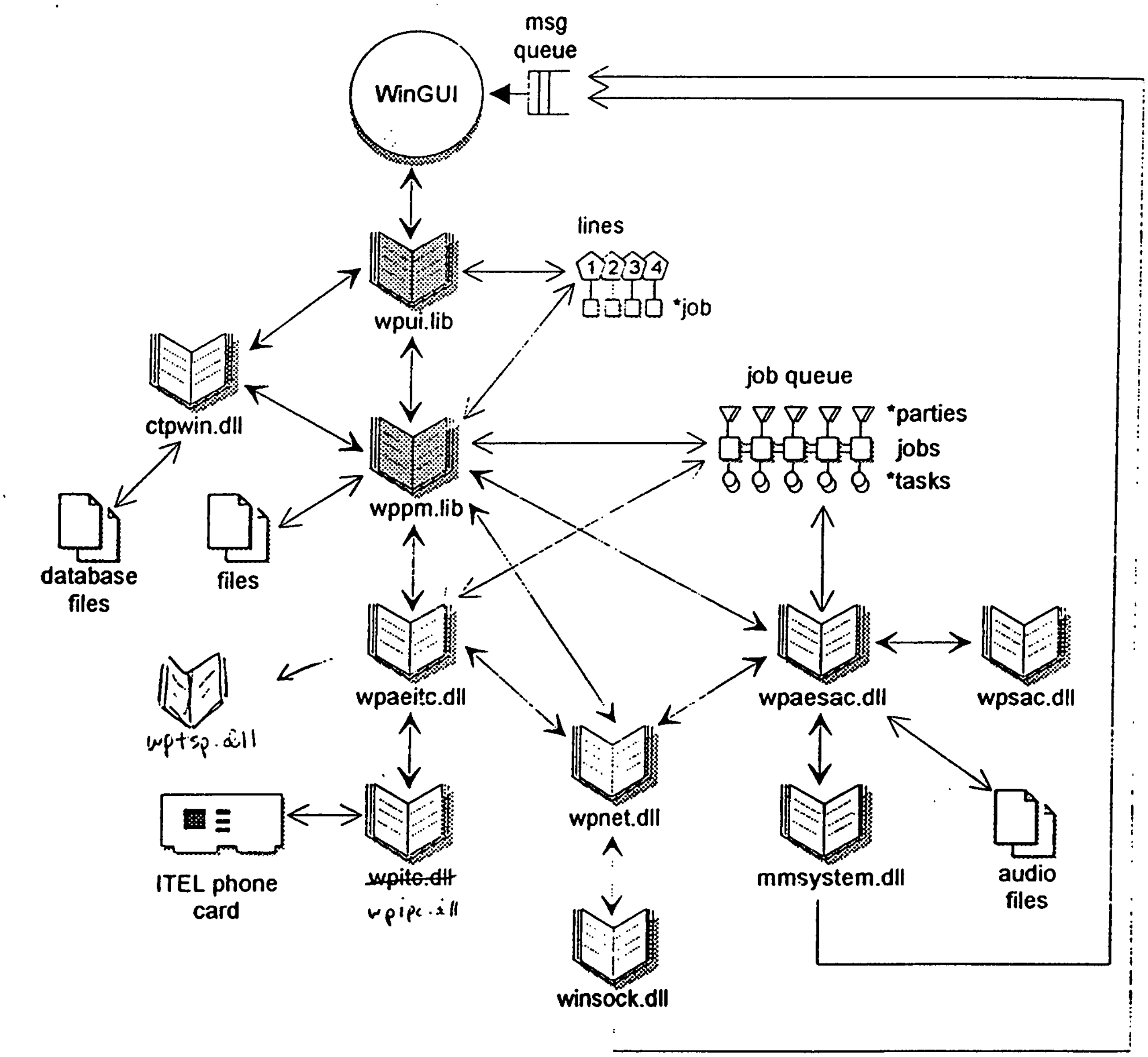


Figure 28

Software Architecture



database	column name	type	Index	comment
webphone.cfg				webphone\webphone.cfg, only 1 record
	state	uchar		0:crippled | 1:registered
	version	char 4		webphone.exe version - n.nn, n=0-9
	dateTime	ulong		installation datetime in secs from 00:00 Jan 1, 1970 GMT
	sndDevice	char 25		name of sound card device driver, null if not found
	ipc	uchar		ITEL Phone Card (IPC) - 0:not found | 1:found
	ipcVersion	char 4		n.nn, n=0-9
	ipcDateTime	ulong		secs from 00:00 Jan 1, 1970 GMT
	voxLevel	ulong		snd card mic voice activiation level setting, units = ?
	firstName	char 10		user's first name
	lastName	char 25		user's last name
	alias	char 10		user's alias or call handle
	emailAddr	char 50		user's e-mail address
	IPaddr	ulong		user's current IP address (assigned if slip/ppp)
	streetAddr	char 50		
	apt	char 5		
	city	char 20		
	state	char 20		
	country	char 20		
	zipcode	char 10		
	phone	char 15		
	fax	char 15		
	company	char 25		
	timezone	uchar		index in TZ array
	infoIPAddr	ulong		IP addr of dir assistance server
	infoHostname	char 20		hostname of dir assistance server
	iconState	uchar		on top when -> 0:never | 1:always | 2:active
	popFrequency	uchar		seconds
	sndCardSpkr	uchar		use snd card as spkr phone -> 0:disabled | 1:enabled
	callBlockAction	uchar		when call blocking enabled -> 0:reject | 1:ansMachine
	fileTransfer	uchar		0:disabled | 1:enabled
	encrypt	uchar		0:disabled | 1:enabled
	ansMachine	uchar		0:disabled | 1:enabled
	timeToAns	uchar		seconds until ans machine picks up
	sndCardVmail	uchar		play vmail on snd card spkr -> 0:disabled | 1:enabled

Spk
mic
value
index 0,1,2,..9

IPaddr
char
~~25~~
50

database	column name	type	index	comment
	ogmNoVmail	uchar		play ogm do not accept vmail -> 0:disabled | 1:enabled
	hearOgm	uchar		hear ogm when played -> 0:disabled | 1:enabled
	noansWav	char 8		filename of wave file to play when no ans
	noansRepeat	uchar		seconds to pause between plays -> 0:no repeat | >0 pause secs
	offlineWav	char 8		filename of wave file to play when offline
	offlineRepeat	uchar		seconds to pause between plays -> 0:no repeat | >0 pause secs
	busyWav	char 8		filename of wave file to play when busy
	busyRepeat	uchar		seconds to pause between plays -> 0:no repeat | >0 pause secs
	dialingWav	char 8		filename of wave file to play when dialing
	dialingRepeat	uchar		seconds to pause between plays -> 0:no repeat | >0 pause secs
	callWav	char 8		filename of wave file to play when call arrives
	callRepeat	uchar		seconds to pause between plays -> 0:no repeat | >0 pause secs
	badAddrWav	char 8		filename of wave file to play when bad email or IP address given
	badAddrRepeat	uchar		seconds to pause between plays -> 0:no repeat | >0 pause secs
	errorWav	char 8		filename of wave file to play when error occurs
	errorRepeat	uchar		seconds to pause between plays -> 0:no repeat | >0 pause secs
	holdWav	char 8		filename of wave file to play when placed on hold
	holdRepeat	uchar		seconds to pause between plays -> 0:no repeat | >0 pause secs
	vmailWav	char 8		filename of wave file to play when vmail arrives
	vmailRepeat	uchar		seconds to pause between plays -> 0:no repeat | >0 pause secs
	priorityWav	char 8		filename of wave file to play when priority ring enabled party calls
	priorityRepeat	uchar		seconds to pause between plays -> 0:no repeat | >0 pause secs
	callackWav	char 8		filename of wave file to play when call acknowledge arrives
	callackRepeat	uchar		seconds to pause between plays -> 0:no repeat | >0 pause secs
	log	uchar		activity log -> 0:disable | 1:enable
	logEvents	uchar		bitmap of events to log -> high nibble = type, low nibble = status
	wpHPos	ulong		saved screen coord for webphone upper left horz pos in pixels
	wpVPos	ulong		saved screen coord for webphone upper left vert pos in pixels
	dirHPos	ulong		saved screen coord for phone dir upper left horz pos in pixels
	dirVPos	ulong		saved screen coord for phone dir upper left vert pos in pixels
	msgHPos	ulong		saved screen coord for vmail msgs upper left horz pos in pixels
	msgVPos	ulong		saved screen coord for vmail msgs upper left vert pos in pixels
	logHPos	ulong		saved screen coord for activity log upper left horz pos in pixels
	logVPos	ulong		saved screen coord for activity log upper left vert pos in pixels
	cfgHPos	ulong		saved screen coord for config upper left horz pos in pixels
	cfgVPos	ulong		saved screen coord for config files upper left vert pos in pixels

database	column name	type	index	comment
	datHPos	ulong		saved screen coord for data files upper left horz pos in pixels
	datVPos	ulong		saved screen coord for data files upper left vert pos in pixels
	iconHPos	ulong		saved screen coord for icon upper left horz pos in pixels
	iconVPos	ulong		saved screen coord for icon upper left vert pos in pixels
	session	ulong		next available session number in sequence
positions of all dialogs	vmailName	ulong		next available vmail name in sequence -> xxxxxxxx, x=0-9
	ogmName	ulong		next available ogm name in sequence -> xxxxxxxx, x=0-9
phonedir.db				webphone\phonedir.db
	number	ulong	key	unique identifier, assigned sequentially
	firstName	char 10		
	lastName	char 25	index	place name if place
	alias	char 10	index	
	emailAddr	char 50		
	IPaddr	ulong	index	
	timezone	uchar		index into TZ array
	type	uchar	index	0:person | 1:place
	priority	uchar	index	0:disable | 1:enable
	blocked	uchar	index	0:disable | 1:enable
	blockAction	uchar		0:REJECT | 1:ACCEPTVMAIL
	ogm Number	ulong	index	link to ogm in ogm.dir
OGM	speedDial	uchar	index	position: 1 - 10, 0:unassigned
1	callFoward	uchar	index	0:disable | 1:enable
	forwardParty	ulong		link to party in phonedir.db
value	index	0: disabled | 1: enable		
	fileBlock	value	'index	∅ : 1
messages.dir				webphone\vmail\messages.dir
	number	ulong	key	unique identifier, assigned sequentially
	direction	uchar	index	0:in | 1:out
	state	uchar	index	0:old | 1:new
	type	uchar		0:gsm | 1:tsp | 2:cripple (play cvmlmsg.wav)

database	column name	type	Index	comment
	filename	char 8		xxxxxxxx.wpm, x=0-9, assigned sequentially
	firstName	char 10		null if place
	lastName	char 25	index	place name if place
	emailAddr	char 50		
	IPaddr	ulong		
	dateTime	ulong	index	secs from 00:00 Jan 1, 1970 GMT
	duration	ulong		secs
files.dir				webphone\files\files.dir
	number	ulong	key	unique identifier, assigned sequentially
	direction	uchar	index	0:in | 1:out
	type	uchar	index	0:executable | 1:email | 2:text | 3:winapp
	filename	char 13		*.ext, ext=exe,bat,sys,txt,doc,wri,xls,pm5,...
	firstName	char 10		null if place
	lastName	char 25	index	place name if place
	emailAddr	char 50		
	IPaddr	ulong		
	dateTime	ulong	index	in or out datetime in secs from 00:00 Jan 1, 1970 GMT
	fileDate	char 8		mm-dd-yy
	fileTime	char 6		hh:mmq, q=a | p
	fileSize	ulong		in bytes
activity.log				webphone\activity.log
	number	ulong	key	unique identifier, assigned sequentially
	firstName	char 10		null if a place
	lastName	char 25	index	
	dateTime	ulong	index	secs from 00:00 Jan 1, 1970 GMT
	emailAddr	char 50		
	IPaddr	ulong		
	type	uchar	index	0:called|1:dialed|2:camped|3:rcv vmail|4:snt vmail|5:rcv file|6:snt file
	status	uchar	index	0:ans|1:noans|2:busy|3:offline|4:success|5:failure|6:disconnect

database	column name	type	index	comment
	vmail	ulong		link to vmail msg in messages.dir
ogm.dir				webphone\ogm\ogm.dir
	number	ulong	key	unique identifier, assigned sequentially
	filename	char 8		xxxxxxxx.wpm, x=0-9, assigned sequentially
	dateTime	ulong		secs from 00:00 Jan 1, 1970 GMT
	description	char 25		
camp.lst				webphone\camp.lst
	number	ulong	key	unique identifier, assigned sequentially
	session	ulong	index	
	direction	uchar	index	0:campee | 1:camper
	dateTime	ulong		
	firstName	char 10		null if a place
	lastName	char 25		
	emailAddr	char 50		
	IPaddr	ulong		

Startup

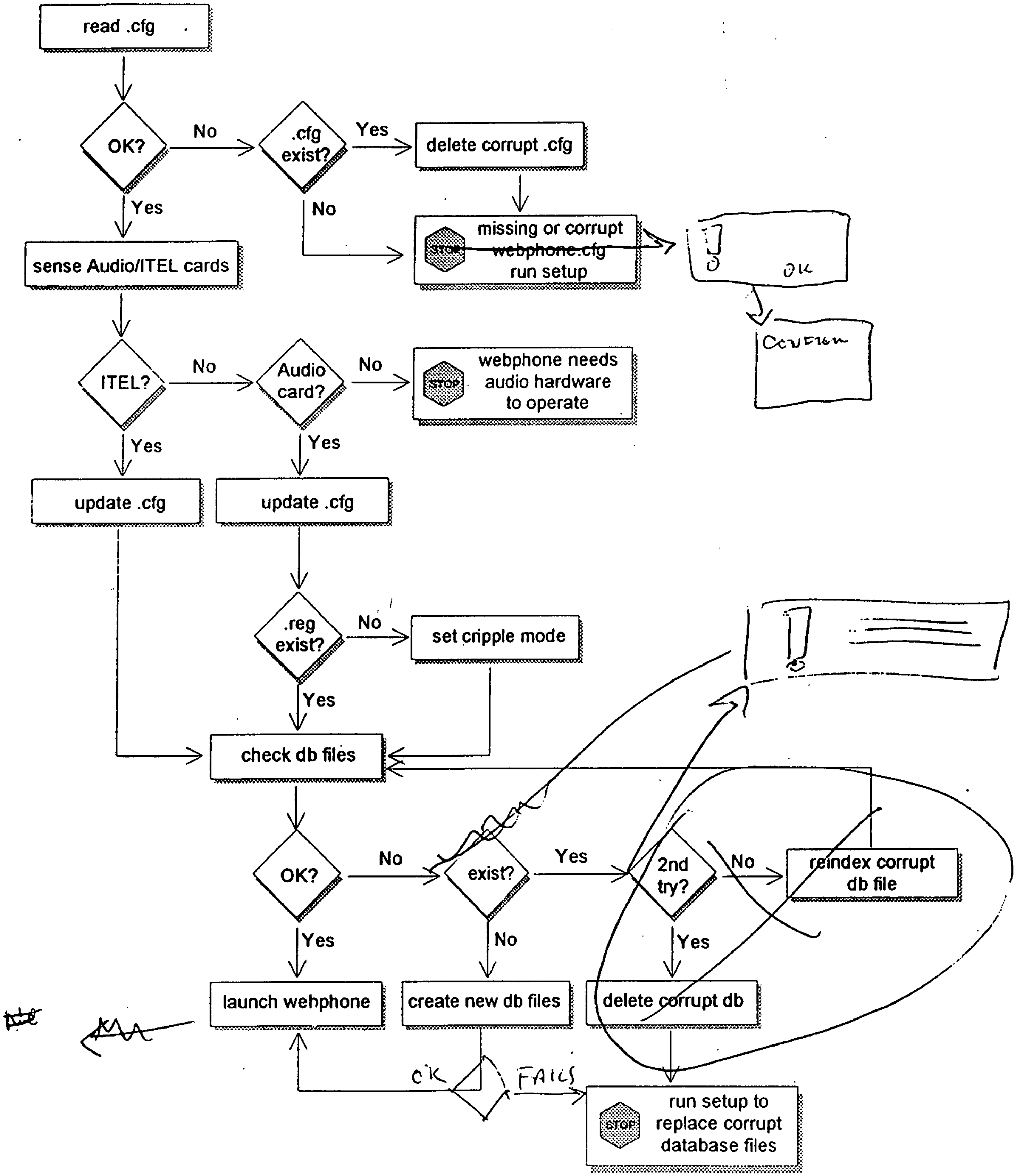
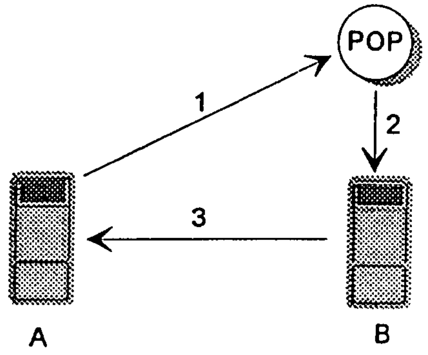


Figure 37

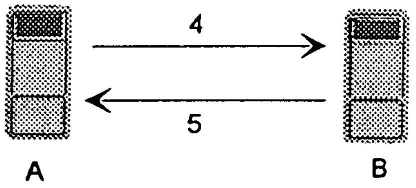
Point to Point Calling Scenario

1. A initiates call to B by sending {CALL}, A says CONNECTING
2. B polls POP and receives {CALL}
3. B xmts <ConnectOK> with B's IP address to A

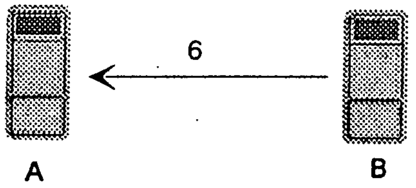


Note: If B's IP address was already known to A then the calling scenario would begin here at step 4:

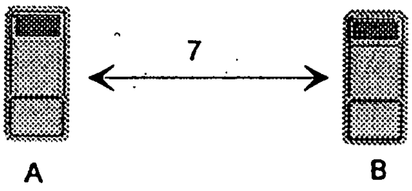
4. A xmts <Call> to B with A's user info
5. B xmts <CallAck> to A with B's user info, A says RINGING, A plays "ringout.wav", B says CALL



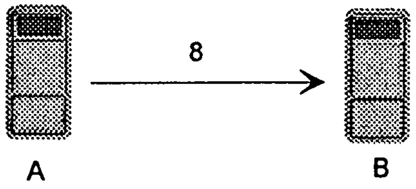
6. when B answers, B xmts <Answer> to A. A stops "ringout.wav" and B stops ringin.wav



7. A and B converse

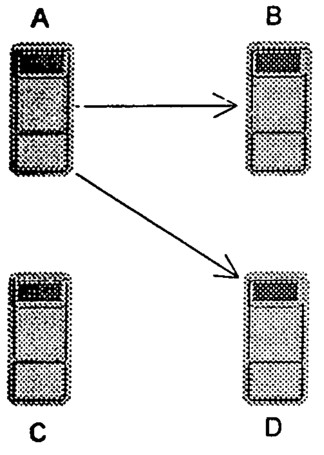


8. A or B presses [END] and xmts <End> to B or A.

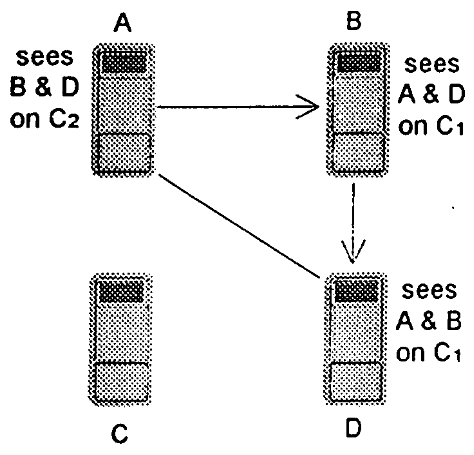


{ } is an e-mail message
< > is a socket message

Conference Calling

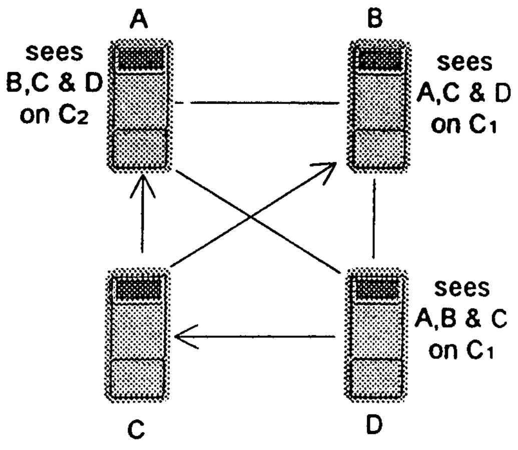


A calls B on L_1 then calls D on L_2



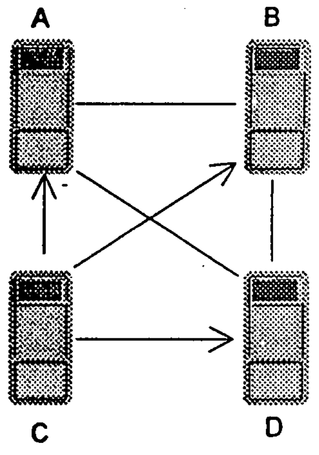
A places B onto L_2 thereby conferencing with B & D.
L_2 then becomes C_2.
A instructs B to call D with <cnfadd>.
B calls D with <cnfcall>.

A xmts to B & D
B xmts to A & D
D xmts to A & B

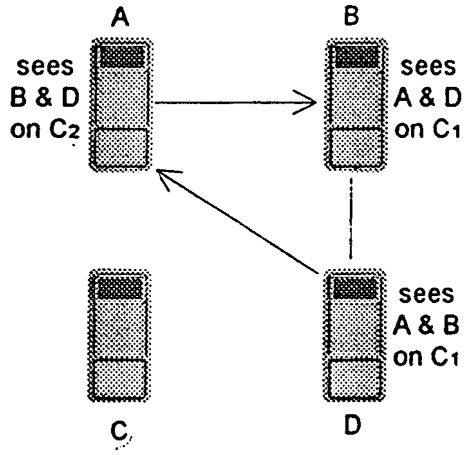


D now calls C and places C on conference. D instructs C to call A & B with <cnfadd>.
C call A & B with <cnfcall>.

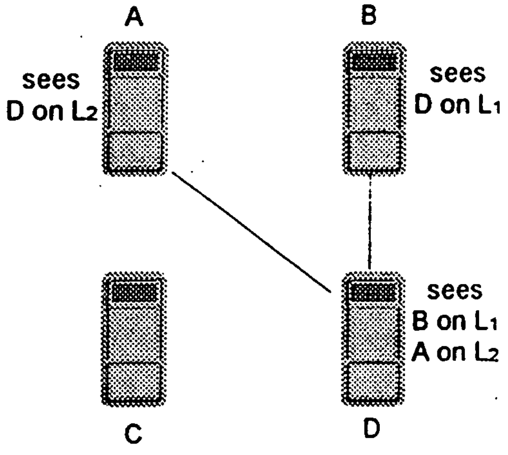
A xmts to B,C & D
B xmts to A,C & D
D xmts to A,B & C



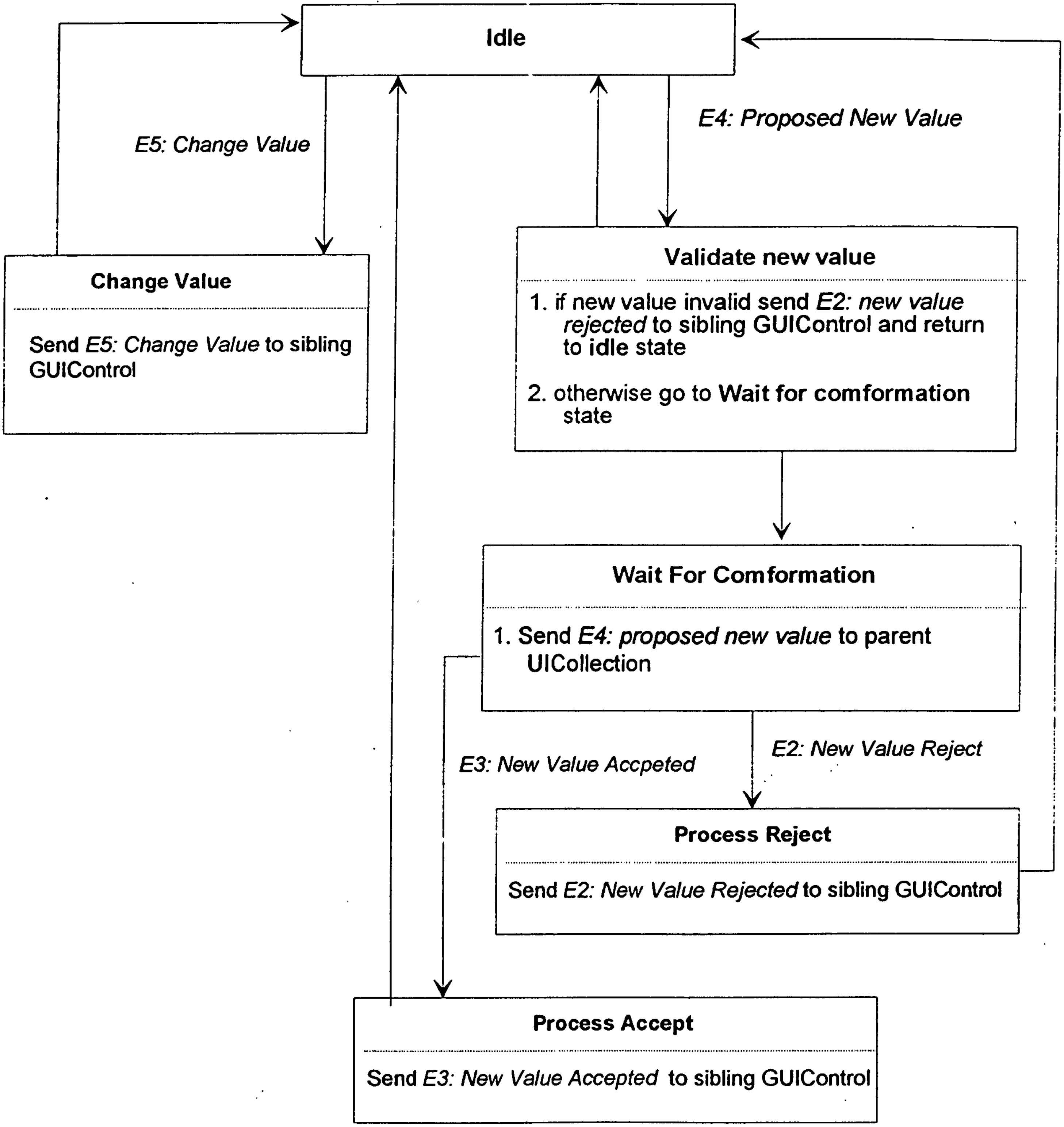
C ends call and sends <end> to A,B & D.



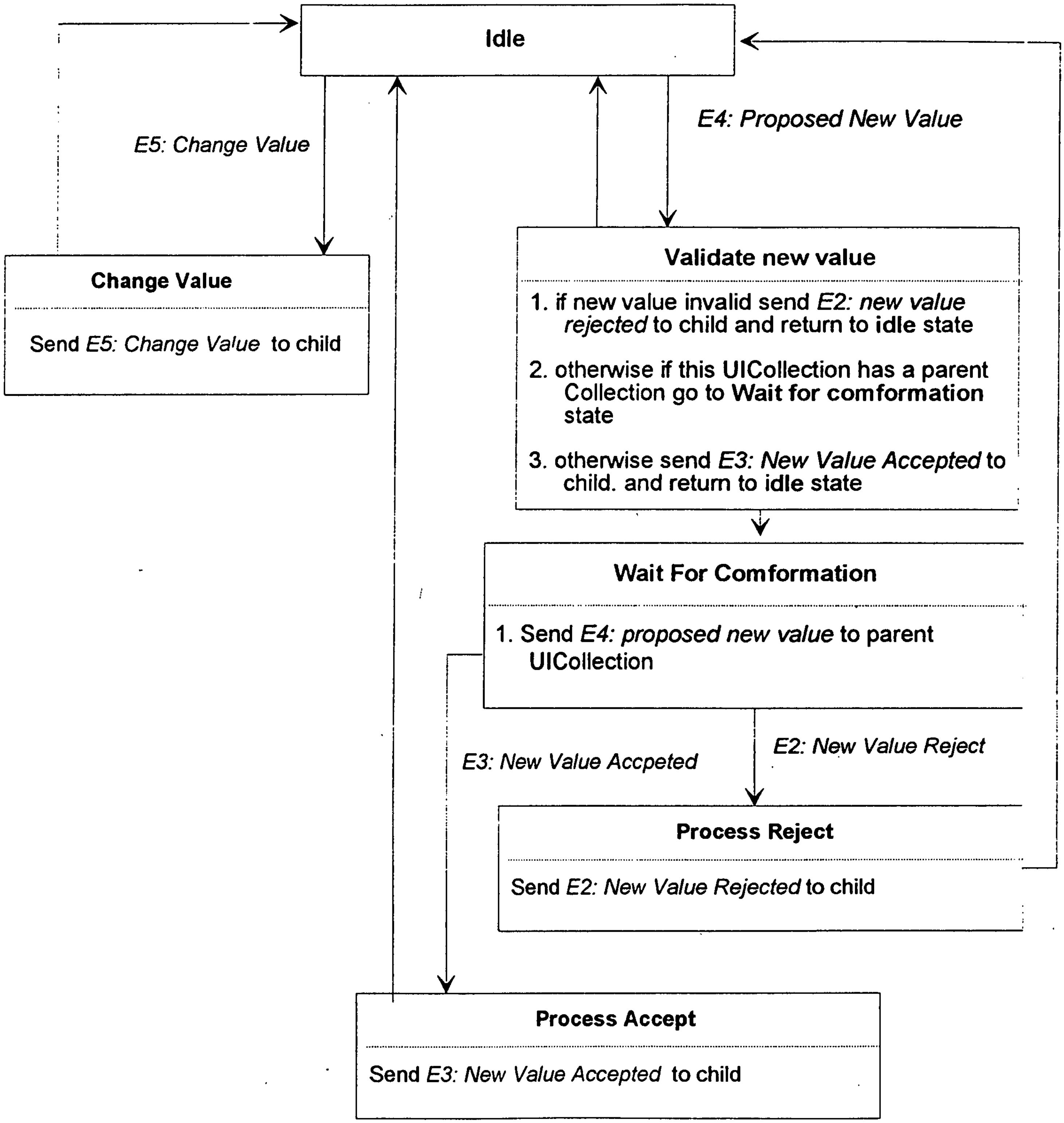
D transfers A onto L_2 thereby dropping A from the conference on C_1. C_1 then becomes L_1.
D sends <cnfdrop> to A.
A then sends <end> to B.



UIControl STD


Idle
E5: Change Value
E4: Proposed New Value
Change Value
Send E5: Change Value to sibling GUIControl
Validate new value
1. if new value invalid send E2: new value rejected to sibling GUIControl and return to idle state
2. otherwise go to Wait for comformation state
Wait For Comformation
1. Send E4: proposed new value to parent UICollection
E3: New Value Accpeted
E2: New Value Reject
Process Reject
Send E2: New Value Rejected to sibling GUIControl
Process Accept
Send E3: New Value Accepted to sibling GUIControl

UICollection STD



Send *E5: Change Value* to child

1. if new value invalid send *E2: new value rejected* to child and return to idle state
2. otherwise if this UICollection has a parent Collection go to **Wait for comformation** state
3. otherwise send *E3: New Value Accepted* to child. and return to idle state

1. Send *E4: proposed new value* to parent UICollection

Send *E2: New Value Rejected* to child

Send *E3: New Value Accepted* to child

Figure 42

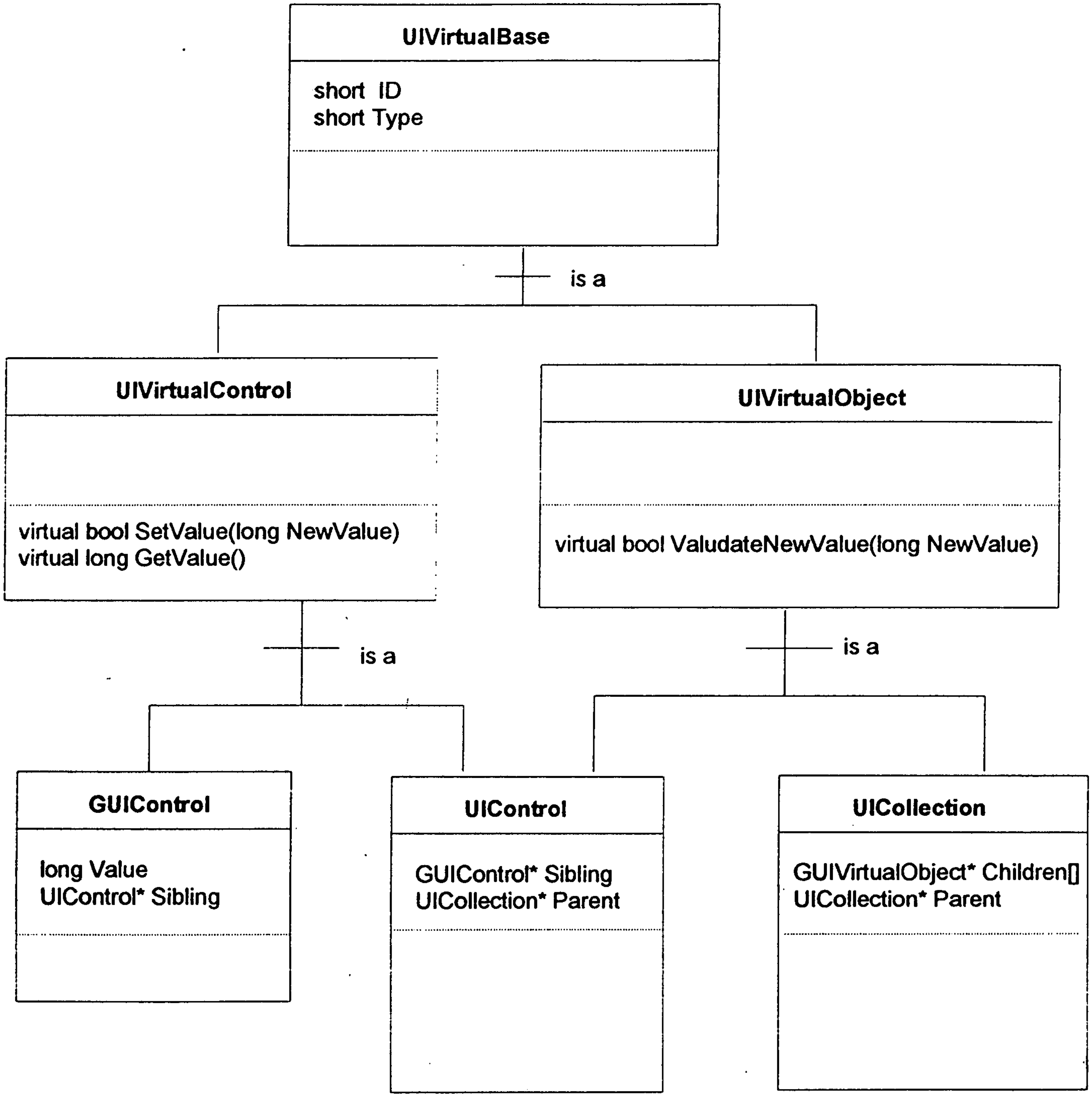

UIVirtualBase
short ID
short Type
is a
UIVirtualControl
virtual bool SetValue(long NewValue)
virtual long GetValue()
UIVirtualObject
virtual bool ValudateNewValue(long NewValue)
is a
is a
GUIControl
long Value
UIControl* Sibling
UIControl
GUIControl* Sibling
UICollection* Parent
UICollection
GUIVirtualObject* Children[]
UICollection* Parent

Figure 43

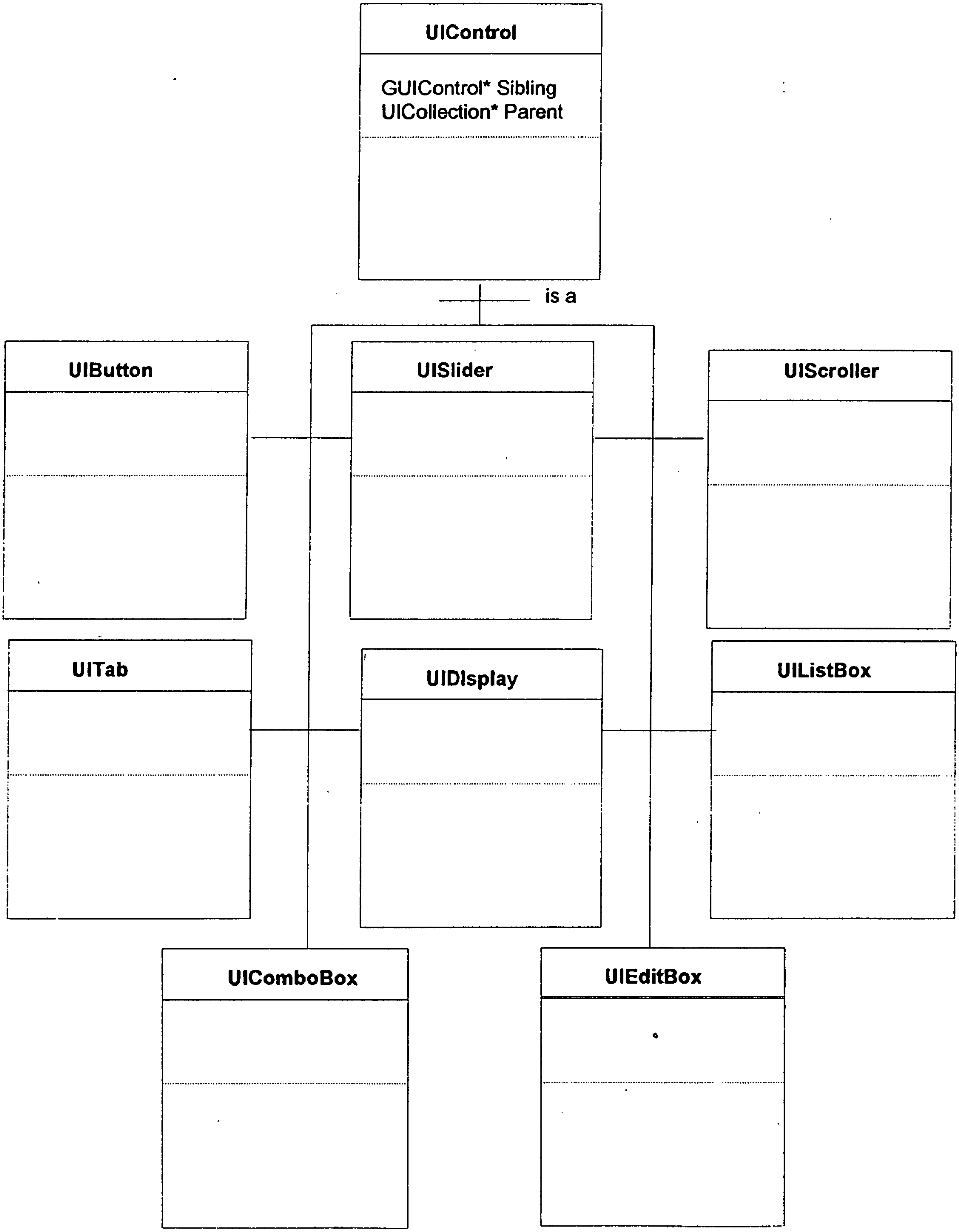

UIControl
GUIControl* Sibling
UICollection* Parent
is a
UIButton
UISlider
UIScroller
UITab
UIDIsplay
UIListBox
UIComboBox
UIEditBox

Figure 44

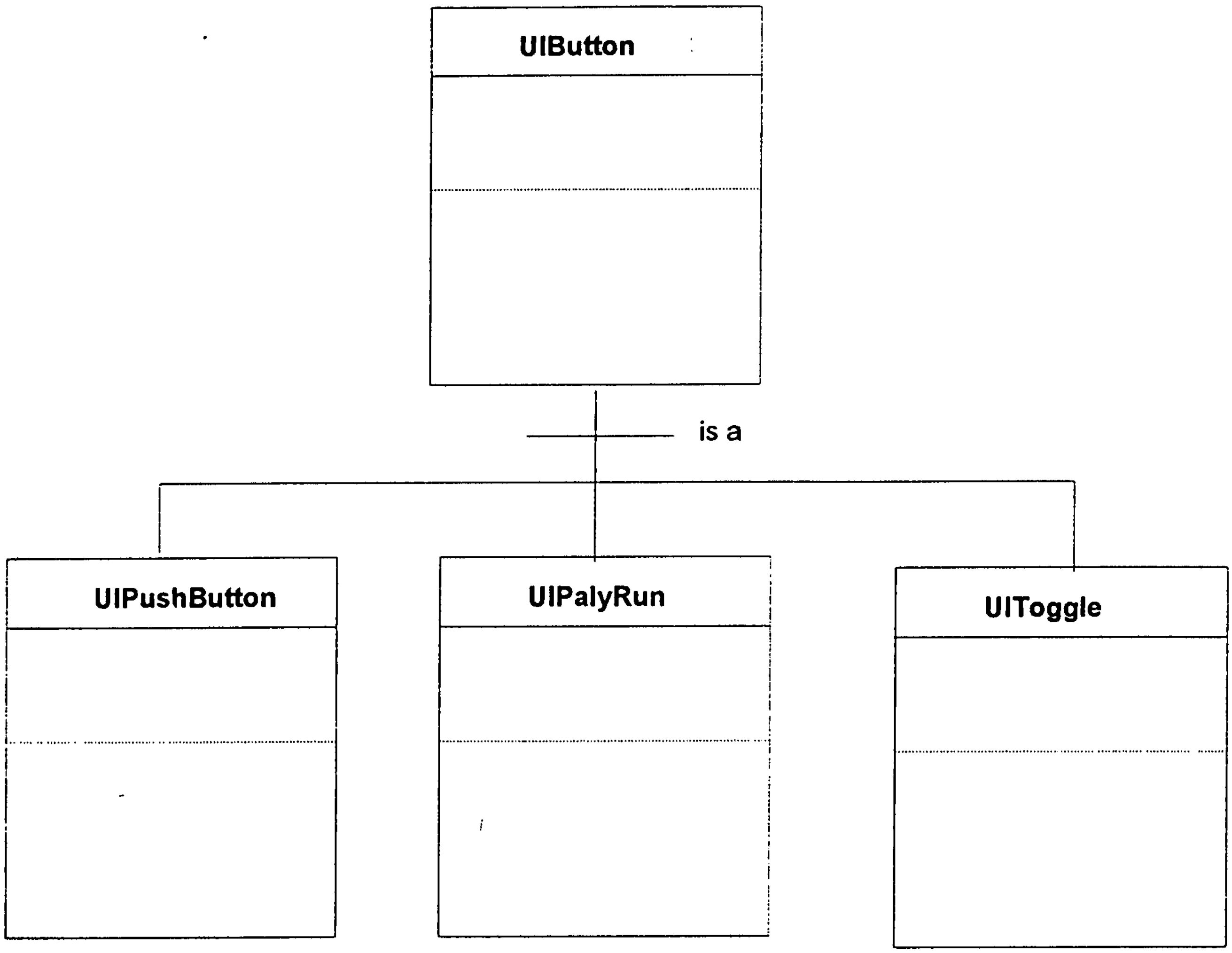


Figure 45

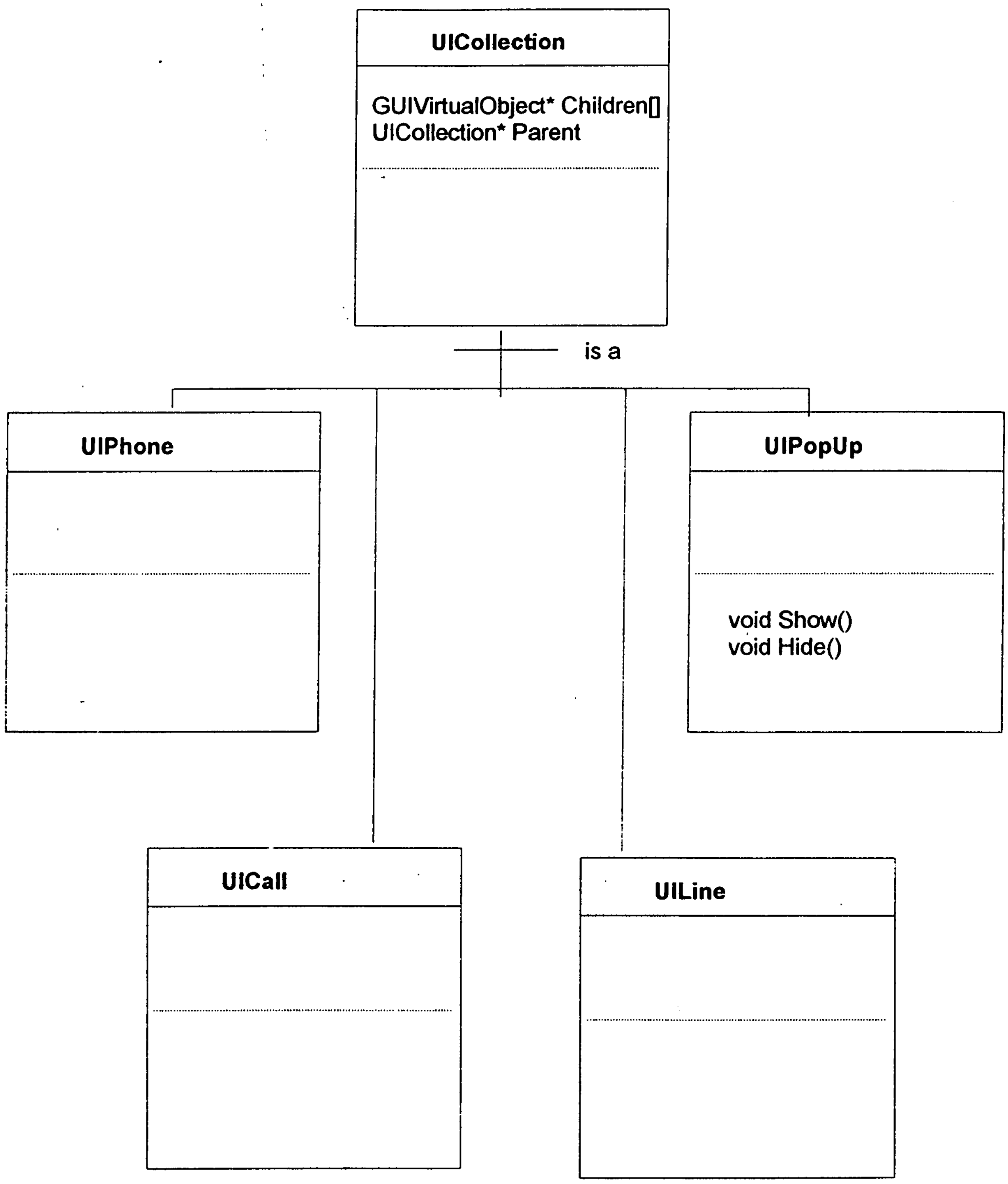

UICollection
GUIVirtualObject* Children[]
UICollection* Parent
is a
UIPhone
UIPopUp
void Show()
void Hide()
UICall
UILine

Figure 46

PhoneManager & AudioEngine Objects

line

⬠

state
duration
*job

CreateCall()
RemoveCall()

flags

Autoans cyn pro

job

□

id
type
state
nparties
*party[ ]
*inTask
*outTask
*nextJob
*prevJob

job()
~job()
AddParty()
RemoveParty()
CreateTask()
RemoveTask()
.
.

party

▽

state
session
socket
partyRec

party()
~party()
LoadParty()
.
.

task

○

cmd
src
dst
state
*job
*buf
extent
fileHandle
fileType
fileLen
fileSize
mic
spkr
flags

task()
~task()
.
.

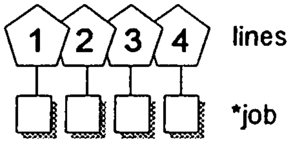
cmd

AE_INIT
AE_CLOSE
AE_START
AE_STOP
AE_FILLME
AE_USEME

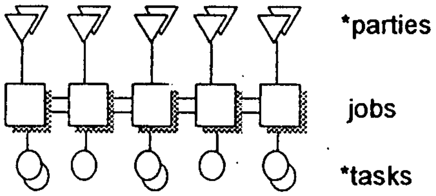
src/dst

AE_MIC
AE_ITCMIC
AE_SPKR
AE_ITCSPKR
AE_SOCKET
AE_FILE

line array



job queue



Job *timers [WP_MAXTIMERS]

index into timer array is TM_?

TM_?

TM_POLL
TM_OHELLO
TM_IHELLO
TM_CALLACK
.
.

socket free list

pre-allocate 16 sockets

Figure 47

Line States

state	value	led color	annunciate
LS_IDLE	0::00000000	gray	IDLE
LS_INUSE	0x00000001	green	INUSE
LS_OFFLINE	0>.00000002	blue	OFFLINE
LS_CONNECTING	0x00000004	blue-green	CONNECTING
LS_CALL	0x00000008	blink green	CALL
LS_RINGOUT	0x00000010	blue-green	RINGING
LS_HOLD	0x00000020	blink red	HOLDING
LS_BUSY	0x00000040	blink blue	BUSY
LS_ANSMACHINE	0x00000080	green	ANSERING MACHINE
LS_REJECTED	0x00000100	blue	REJECTED
LS_DISCONNECTED	0x00000200	black	DISCONNECTED
LS_NETFAILURE	0x00000400	black	NETWORK FAILURE
LS_COMMFAILURE	0x00000800	black	COMMUNICATIONS FAILURE
LS_CAMPACK	0x00001000	blink blue-green	PARTY AVAILABLE
LS_OGMPLAY	0x00002000	blink green	PLAYING MESSAGE
LS_VMAILRCV	0x00004000	blink green	RECEIVING VOICE MAIL
LS_RECORD	0x00008000	red	RECORDING
LS_PLAY	0x00010000	orange-yellow	PLAYING
LS_SELECT	0x01000000	gray	IDLE
LS_MUTE	0x02000000	yellow	MUTE
LS_ONHOLD	0x04000000	red	ONHOLD

LS_TALK	0X08000000	green	TALK
LS_LISTEN	0X10000000	green	LISTEN

Job States

state
JS_IDLE
JS_DONE
JS_SELECT
JS_OCALL
JS_ORING
JS_IRING
JS_ICONNECT
JS_OCONNECT
JS_ERROR
JS_OFFLINE
JS_BUSY
JS_RBUSY
JS_INUSE
JS_DISCONNECTED
JS_HOLD
JS_ONHOLD
JS_HOLDNONHOLD
JS_ORINGHOLD
JS_OCALLHOLD
JS_RBUSYHOLD
JS_OCONNECTHOLD
JS_OGMRCV
JS_OVMAILRECWAIT
JS_OVMAILREC
JS_OVMAILXMT
JS_OVMAILPLAY
JS_OVMAILPAUSE
JS_OGMPLAY
JS_IVMAILRECWAIT
JS_IVMAILREC
JS_CAMPACK
JS_FILEXMT
JS_FILEXMTACK
JS_FILERCV
JS_FILESND
JS_EMAILRCV
JS_INFOACK
JS_INFORCV
JS_OGMPAUSE
JS_OGMPLAY
JS_OGMREC
JS_VMAILPLAY
JS_VMAILPAUSE
JS_VMAILRCV
JS_EMAILFILERCV
JS_USERINFO

Figure 49

User Interface Events

<table>
<tr><th>Action</th><th>GUI event</th><th>PM event</th></tr>
<tr><td>open phone directory</td><td>press [DIR]</td><td></td></tr>
<tr><td>open voice mail messages dialog</td><td>press [MSG]</td><td></td></tr>
<tr><td>open activity log</td><td>press [LOG]</td><td></td></tr>
<tr><td>open configuration control dialog</td><td>press [CFG]</td><td></td></tr>
<tr><td>open data files dialog</td><td>press [DAT]</td><td></td></tr>
<tr><td>open help system</td><td>press [?] | press ? in dialogs</td><td></td></tr>
<tr><td>display bubble help</td><td>point to any [ ]|[ ]v for 1.5 seconds</td><td></td></tr>
<tr><td>display party information</td><td>rt clk on [n]|[Li]|[Lh]</td><td></td></tr>
<tr><td>display camp list</td><td>rt clk on [CMP]</td><td></td></tr>
<tr><td>display call block list</td><td>rt clk on [BLK]|[BLK]v</td><td></td></tr>
<tr><td>display priority ring list</td><td>rt clk on [PRI]|[PRI]v</td><td></td></tr>
<tr><td>display conference list</td><td>rt clk on [Ci]|[Ch]; press dn arrow in display</td><td></td></tr>
<tr><td>save voice mail to file system</td><td>drag selected voice mail to dir in WFM</td><td></td></tr>
<tr><td>save ogm to file system</td><td>drag selected ogms to dir in WFM</td><td></td></tr>
<tr><td>restore voice mail from file system</td><td>drag selected .wpm files to Voice Mail dialg</td><td></td></tr>
<tr><td>restore or add ogm from file system</td><td>drag selected ogms from dir in WFM to OGM dialog</td><td></td></tr>
<tr><td>add party on line to phone directory</td><td>press [DIR]; drag [Li]|[Lh] to DIR</td><td></td></tr>
<tr><td>add party on conf line to phone directory</td><td>rt clk [Ci]|[Ch]; press dn arrow; drag party to DIR</td><td></td></tr>
<tr><td>assign party to speed dial</td><td>press [DIR];drag party to [n] where n != .</td><td></td></tr>
<tr><td>"</td><td>drag [Li]|[Lh] to [n] where n != .</td><td></td></tr>
<tr><td>place an IP based call</td><td>press [n];[n];...[n];[SND]</td><td>PM_IPCALL</td></tr>
<tr><td>place an e-mail or IP based call</td><td>name;[SND]</td><td>PM_CALL | PM_IPCALL</td></tr>
<tr><td>"</td><td>drag Party from DIR to [Lf]</td><td>"</td></tr>
<tr><td>"</td><td>press [DIR];dbl clk on party in DIR</td><td>"</td></tr>
<tr><td>recall the last party called</td><td>press [RCL]</td><td>"</td></tr>
<tr><td>"</td><td>drag [RCL] to [Lf]</td><td>"</td></tr>
<tr><td>speed dial</td><td>drag [n] to [Lf]</td><td>"</td></tr>
<tr><td>"</td><td>press [n];[SND]</td><td>"</td></tr>
<tr><td>call party from activity log</td><td>press [LOG]; dbl clk on log entry</td><td>"</td></tr>
<tr><td>"</td><td>drag log entry to [Lf]</td><td>"</td></tr>
</table>

User Interface Events

Action	GUI event	PM event
answer a call	press [SND]	PM_ANSWER
"	press [Lc]	"
pre-select a line	press [Lf]	PM_SELECT, ON
deselect a line	press [Ls]	PM_SELECT, OFF
place a call on hold	press [Li]|[Ci]	PM_HOLD, ON
"	press [HLD]	"
"	press [Lx]|[Cx] where Lx != Li | Cx != Ci	"
"	press [RCL]	"
take a call off hold	press [Lh]|[Ch]	PM_HOLD, OFF
end a call	press [END]	PM_END
mute a line	press [MUT]	PM_MUTE, ON
take mute off a line	press [MUT]v	PM_MUTE, OFF
enable call blocking	press [BLK]	PM_BLK, ON
disable call blocking	press [BLK]v	PM_BLK, OFF
add party to call block list	update party in DIR | drag party to [BLK]	PM_UPDBLK,,, ADD
delete party from call block list	remove party from block list in display	PM_UPDBLK,,,DELETE
enable do not disturb	press [DND]	PM_DND, ON
disable do not disturb	press [DND]v	PM_DND, OFF
enable priority ringing	press [PRI]	PM_PRI, ON
disable priority ringing	press [PRI]v	PM_PRI, OFF
add party to priority ring list	update party in DIR	PM_UPDPRI,,, ADD
delete party from priority ring list	remove party from priority ring list in display	PM_UPDPRI,,,DELETE
camp on a busy or offline call	press [CMP]	PM_CAMP, line
remove camp on party	rt clk on [CMP]; delete party from camp list	
enable call forwarding	press [FWD]	PM_FWD, ON
disable call forwarding	press [FWD]v	PM_FWD, OFF
assign party to call forward	drag party in DIR|[Li]|[Lh]|[n] to [FWD]	PM_FWD, *party
transfer a party to another line	drag [Li]|[Lh] to a [Lf]	PM_LINEXFR
add on or more parties to conference	drag [Li]|[Lh]|[Ci]|[Ch] to another [Li]|[Lh]|[Ci]|[Ch]	PM_CNFADD (for each party)
transfer a party from a conf to a line	drag party from conf list to [Lf]	PM_CNFDROP
transfer a party from one conf to another	drag party from conf list to another [Ci]|[Ch]	PM_CNFDROP ; PM_CNFADD
remove a party from a conference	select party in conf list and press [END]	PM_CNFDROP
start recording audio	press [*]	PM_ACREC
start playing audio	press [|>]	PM_ACPLAY

User Interface Events

Action	GUI event	PM event
stop rec or playing audio	press [stop]	PM_ACSTOP
pause rec or playing audio	press [||]	PM_ACPAUSE
rewind audio to beginning	press [|<]	PM_ACRWD
fast forward audio to end	press [>|]	PM_ACFWD
cancel audio record session	press [x]	PM_ACABORT
finished recording voice mail	press [END]	PM_ACEND
finished recording ogm	select another ogm	"
play audio file to party on line	drag vmail from MSG to [Li]|[Lh]|[Ci]|[Ch]	PM_ACPLAY
"	drag ogm from OGM to [Li]|[Lh]|[Ci]|[Ch]	"
"	drag audio file from WFM to [Li]|[Lh]|[Ci]|[Ch]	"
transfer file(s) to one or more parties	drag file(s) from WFM to [Li]|[Lh]|[Ci]|[Ch]	PM_FILEXFR
"	drag file(s) from WFM to selected parties in DIR	"
abort file transfers	press [DAT]; select file in *Data Files Out*; press [x]	PM_FILEXFRABORT
request directory assistance	bad name;[SND] | press [DIR];press [Info]	PM_INFOREQ
abort directory assistance request	press [x] in Information dialog	PM_INFOABORT

Key to symbols

[ ] = button is up
[ ]v = button is down
n = 0,1,2,3,4,5,6,7,8,9, .
L = single line (1 party)
C = conference line (> 1 party)
Lf = free line
Lc = call on line
Lh = hold on line
Li = in use line
Ls = selected line
; = then
| = or
WFM = MS Windows File Manager

~~PM_PLAYFILE *job~~ ● ~~*file~~ (File →Socket

PM_SELECTFILE *job ~~*~~file (initJob so wav file is associated)

UI Triggered PM Events

event	arg1	arg2	arg3	comment
PM_INIT				
PM_CLOSE				
PM_CALL	*job	lineID		initiate email call
PM_IPCALL	*job	lineID		initiate IP call
PM_ANSWER		lineID		answer received
PM_HOLD		lineID	ON | OFF	toggle hold
PM_SELECT		lineID	ON | OFF	toggle line selection
PM_END		lineID		end call
PM_MUTE		lineID	ON | OFF	toggle muting
PM_BLK			ON | OFF	toggle call block
PM_UPDBLK	*party		ADD | DELETE	add or del party from blk list
PM_DND			ON | OFF	toggle do not disturb
PM_FWD	*party		ON | OFF	arg1 or arg3
PM_CAMP	sid	lineID		sid - OFF | lineID - ON
PM_PRI			ON | OFF	toggle priority ringing
PM_UPDPRI	*party		ADD | DELETE	add or del party to priority ring list
PM_LINEXFR		lineID	lineID	lineIDs not the same
PM_CNFADD	partyID	lineID		add party to cnf
PM_CNFDROP	partyID	lineID		drop party from cnf
PM_PARTYXFR	partyID	lineID	lineID	lineIDs not the same
PM_INFOREQ	*job	char*		directory assistance request
PM_INFOABORT	*job			abort directory assistance
PM_FILEXFR	*job	char *		initiate file transfer
PM_FILEXFRABORT	*job			abort file transfer
PM_ACSTOP	*job			audio control stop
PM_ACPLAY	*job			audio control play
PM_ACPAUSE	*job			audio control pause
PM_ACREC	*job			audio control record
PM_ACABORT	*job			audio control cancel
PM_ACRWD	*job			audio control rewind
PM_ACFWD	*job			audio control forward
PM_ACEND	*job			psuedo-control: lose focus
PM_MIC	*job	*buf		microphone I/O
PM_SPKR	*job	*buf		speaker I/O
PM_SOCKET	*job	*buf	READ | WRITE	socket I/O
PM_TIMEOUT	*job		TM_?	timer elapsed

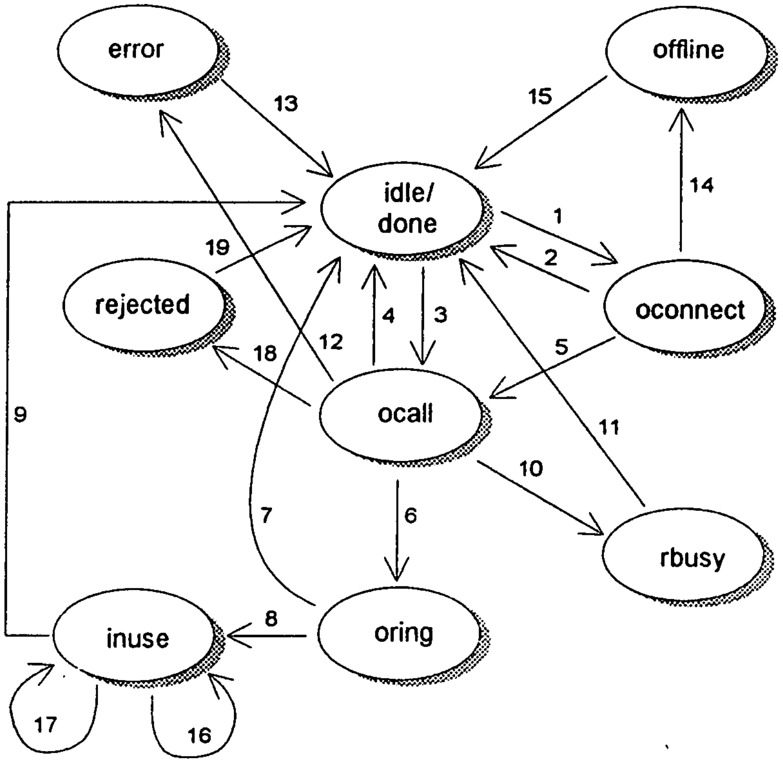
— vol

inforeq "record" search

first \0
last \0
company \0
city \0
state \0
county \0

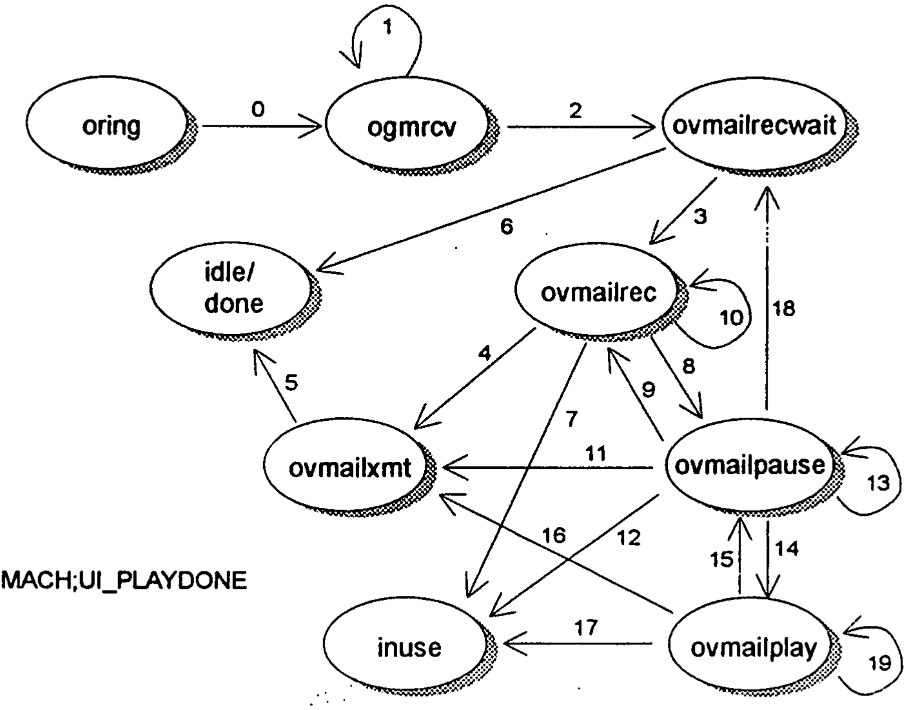
Figure 53

PhoneManager State-Event Diagrams



Placing a call events

1. PM_CALL ; {CALL} –>
2. PM_END
3. PM_IPCALL | <– {CAMPCALL} ; <Call> –>
4. PM_END
5. <– <ConnectOK> ; <Call> –>
6. <– <CallAck>;LS_RINGOUT;UI_CALLACK
7. PM_END
8. <– <Answer>;LS_INUSE;UI_CALLANSWER
9. PM_END | <– <End>;LS_DONE;UI_CALLEND
10. <– <Busy>;LS_BUSY;UI_CALLBUSY
11. PM_END
12. PM_TIMEOUT;LS_COMMFAIL;UI_COMMFAIL
13. PM_END
14. PM_TIMEOUT;LS_OFFLINE;UI_OFFLINE
15. PM_END
16. PM_SOCKET
17. PM_MIC | PM_SPKR
18. <– <Reject>;LS_REJECTED;UI_CALLREJECTED
19. PM_END

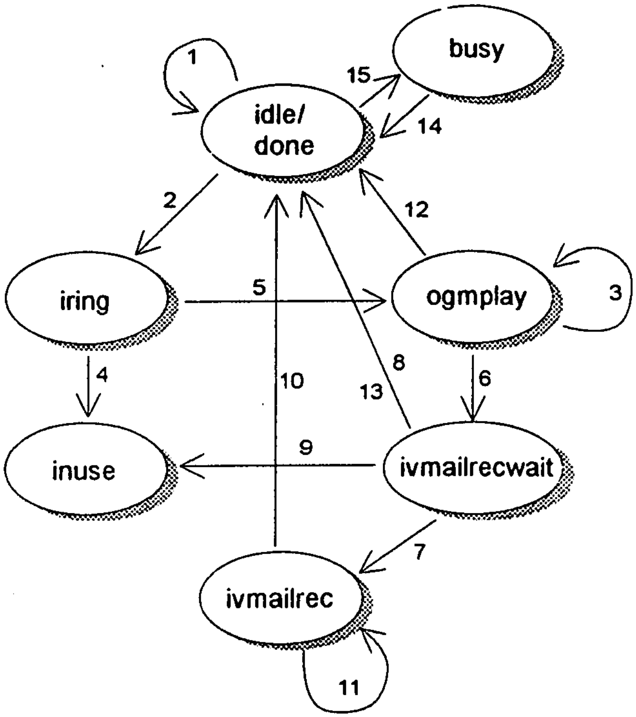


Recording and sending vmail Events

0. <– <AnsMachine>;LS_ANSMACH;UI_ANSMACH
1. PM_SOCKET
2. <– <OgmEnd>;UI_VMAILREC
3. PM_ACREC
4. PM_END
5. end of file;UI_VMAILSENT
6. PM_END
7. <– <Answer>;LS_INUSE;UI_CALLANSWER
8. PM_ACPAUSE | PM_ACSTOP
9. PM_ACREC
10. PM_MIC;AE_USEME;UI_AUDIOSTS
11. PM_END
12. <– <Answer>;LS_INUSE;UI_CALLANSWER
13. PM_ACRWD | PM_ACFWD
14. PM_ACPLAY
15. PM_ACPAUSE | PM_ACSTOP | end of file;LS_ANSMACH;UI_PLAYDONE
16. PM_END
17. <– <Answer>;LS_INUSE;UI_CALLANSWER
18. PM_ACABORT
19. PM_SPKR;AE_FILLME;UI_AUDIOSTS

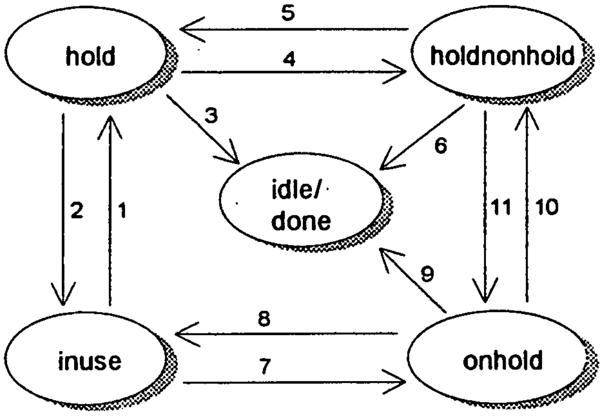
Figure 54

PhoneManager State-Event Diagrams



Inbound call and answering machine events

1. <-- {CALL} ; <ConnectOK> -->
2. <-- <Call> ; <CallAck> -->;LS_CALL;UI_CALL
3. <Audio> -->
4. PM_ANSWER ; <Answer> -->
5. PM_TIMEOUT ; <AnsMachine> -->;LS_OGMPLAY;UI_OGMPLAY
6. end of file ; <OgmEnd> -->;LS_VMAILRCV;UI_VMAILRCV
7. <-- <Audio> ; AE_START;LS_DONE;UI_CALLEND
8. <-- <End>;LS_DONE;UI_CALLEND
9. PM_ANSWER ; <Answer> -->
10. <-- <End> ; AE_STOP;LS_DONE;UI_VMAILRCVD
11. PM_SOCKET;AE_USEME
12. <-- <End>;LS_DONE;UI_CALLEND
13. <-- <Camp>;UI_CAMPRCV
14. <-- <End>|PM_TIMEOUT|<-- <Camp>;UI_CAMPRCV
15. <-- <Call> w/ NO AVAIL LINES & ANSMACH disabled



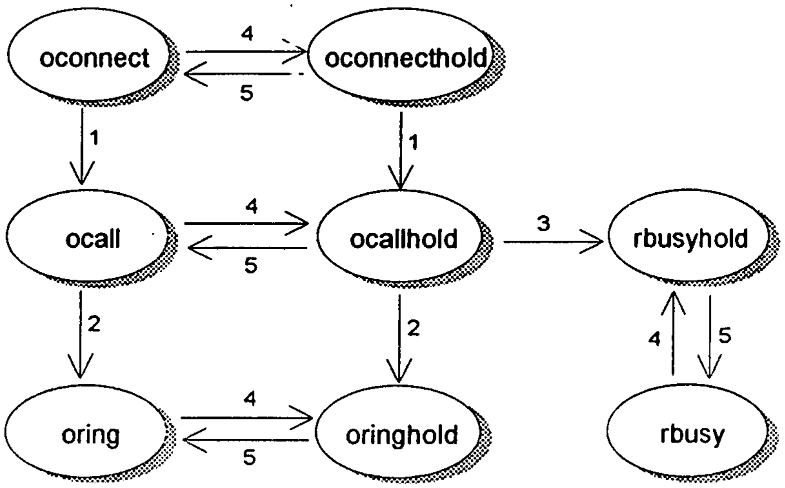
Hold events

1. PM_HOLD, ON
2. PM_HOLD, OFF
3. PM_END ; <-- <End>;LS_DONE;UI_CALLEND
4. <-- <Hold, ON>;LS_ONHOLD;UI_ONHOLD
5. <-- <Hold, OFF>;LS_OFFHOLD;UI_OFFHOLD
6. PM_END, <-- <End>LS_DONE;UI_CALLEND
7. <-- <Hold, ON>;LS_ONHOLD;UI_ONHOLD
8. <-- <Hold, OFF>;LS_OFFHOLD;UI_OFFHOLD
9. PM_END, <-- <End>;LS_DONE;UI_CALLEND
10. PM_HOLD, ON
11. PM_HOLD, OFF

PhoneManager State-Event Diagrams

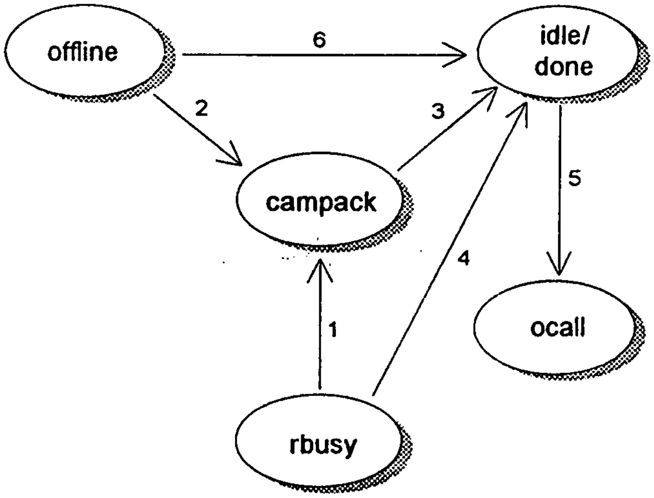
More hold events

1. <-- <ConnectOK> ; <Call> -->
2. <-- <CallAck>;UI_CALLACK
3. <-- <Busy>;LS_RBUSY;UI_CALLBUSY
4. PM_HOLD, ON
5. PM_HOLD, OFF



Camping events

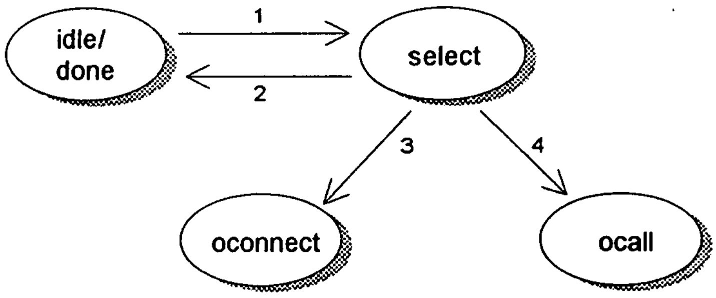
1. PM_CAMP,line ; <Camp> -->
2. PM_CAMP, line ; {CAMPCALL} -->
3. <-- <CampAck> ; LS_CAMPACK;UI_CAMPACK
4. PM_END; <End> -->
5. PM_IPCALL ; <Call> -->
6. PM_END



PhoneManager State-Event Diagrams

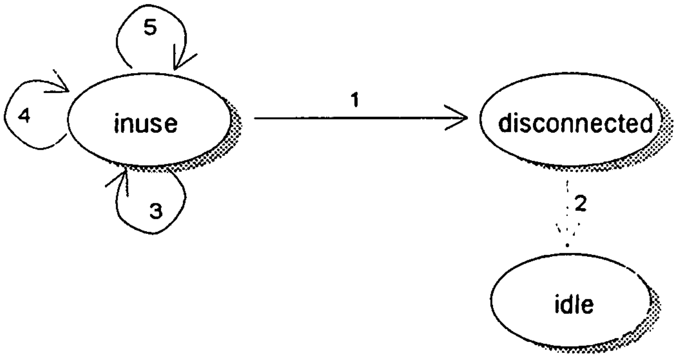
Select events

1. PM_SELECT, ON
2. PM_SELECT, OFF
3. PM_CALL
4. PM_IPCALL



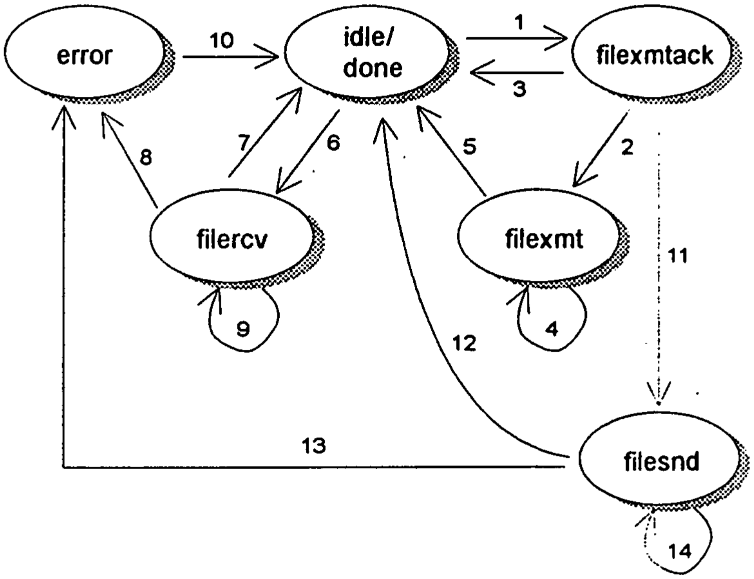
Hello events

1. PM_TIMEOUT, ihello
2. PM_END
3. <-- <Hello> | PM_SOCKET, READ ; TM_IHELLO
4. PM_TIMEOUT, ohello ; <Hello> -->
5. PM_MIC | PM_SOCKET, WRITE ; TM_OHELLO



File transfer events

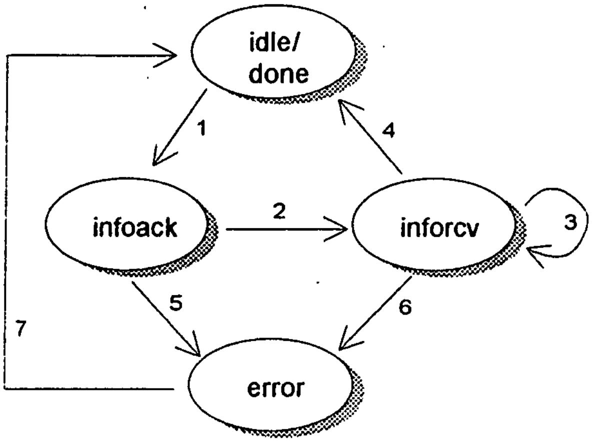
1. PM_FILEXFR ; <FileXmtReq> -->
2. <-- <FileXmtAck>
3. PM_FILEXFRABORT;<Filexfrabort> -->
4. <File> -->;UI_FILEXFRSTS
5. end of file ; <FileXmtEnd> -->UI_FILEXFRSTS
6. <-- <FileXmtReq> ; <FileXmtAck> -->
7. <-- <FileXmtAbort> | <-- <FileXmtEnd>;UI_FILEXFREND
8. PM_TIMEOUT, file
9. <-- <File>
10. job.state = LS_ERROR;UI_FILEXFRFAILURE
11. PM_TIMEOUT, filexmtack
12. end of file;UI_FILEXFREND
13. failure to email
14. {FILEXFR} -->;UI_FILEXFRSTS



PhoneManager State-Event Diagrams

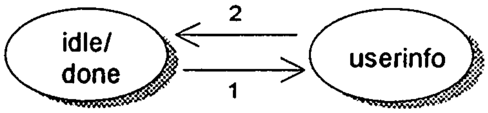
Dire:tory assistance events

1. PM_INFOREQ; <InfoReq> –>
2. <– <InfoAck>;UI_INFOACK
3. <– <Info>;UI_INFO
4. <– <InfoEnd>;UI_INFOEND
5. PM_TIMEOUT, infoack;UI_INFOFAILURE
6. PM_TIMEOUT, info;UI_INFOFAILURE
7. job.state = LS_ERROR



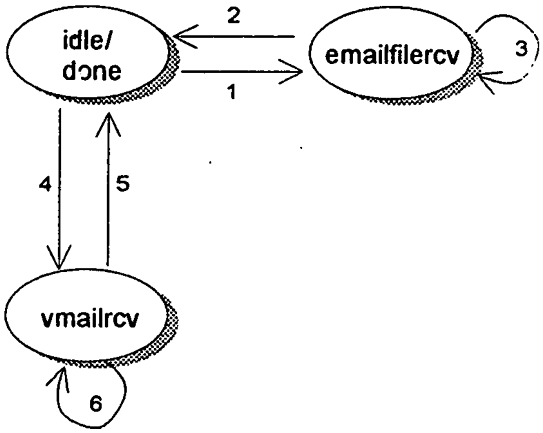
Operator initiated user info acquisition

1. <– <Userinforeq>
2. <Userinfo> –>



Receive Vmail, Email & Files via POP

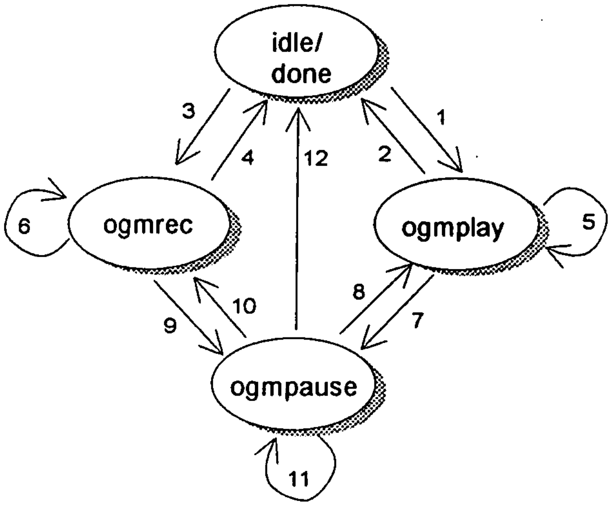
1. <– {EMAIL} | <– {FILEXFR}
2. end of file;UI_FILEXFREND
3. read chunk-o-file;UI_FILEXFRSTS
4. <– {VMAIL}
5. end of file;UI_VMAILRCVD
6. read chunk-o-vmail



PhoneManager State-Event Diagrams

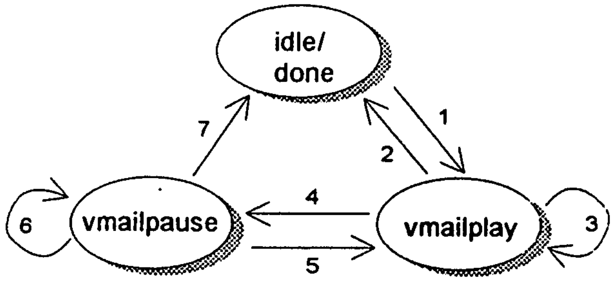
Recording and playing OGMs

1. PM_ACPLAY
2. PM_ACABORT
3. PM_ACREC
4. PM_ACABORT
5. PM_SPKR;AE_FILLME;UI_AUDIOSTS
6. PM_MIC;AE_USEME;UI_AUDIOSTS
7. PM_ACPAUSE|PM_ACSTOP|end of file
8. PM_ACPLAY
9. PM_ACPAUSE|PM_ACSTOP|rec file full
10. PM_ACREC
11. PM_ACRWD | PM_ACFWD
12. PM_ACABORT



Playing Vmail

1. PM_ACPLAY
2. PM_ACEND | lost focus
3. PM_SPKR;AE_FILLME;UI_AUDIOSTS
4. PM_ACPAUSE | PM_ACSTOP
5. PM_PLAY
6. PM_ACRWD | PM_ACFWD
7. PM_ACEND | lost focus



PM triggered UI Events

event	arg1	arg2	UI actions
UI_CAMPACK	lineID		LED:blink blue-green;play *campack.wav*
UI_CAMPRCV		*job	upd camp.lst;upd camp list memory image
UI_CALL	lineID		annunc:"CALL";play *ringin.wav;* chg LED
UI_CALLACK	lineID		annunc:"RINGING";play *ringout.wav;* chg LED
UI_CALLANSWER	lineID		stop play: annunc:"INUSE"; chg LED
UI_CALLEND	lineID		annunc:"IDLE";chg LED
UI_CALLBUSY	lineID		annunc:"BUSY";play *busy.wav;* chg LED
UI_CALLOFFLINE	lineID		stop play;annunc:"OFFLINE";chg LED
UI_CALLOHHOLD	lineID		annunc:"ON HOLD";chg LED
UI_CALLOFFHOLD	lineID		annunc:"IN USE";chg LED
UI_COMMFAIL	lineID		annunc:"COMMUNICATIONS FAILURE";chg LED
UI_CALLREJECT	lineID		annunc:"CALL REJECTED";chg LED
UI_ANSMACHINE	lineID		annunc:"ANSWERING MACHINE"
UI_VMAILREC	lineID		activate audio controls
UI_VMAILSENT		*job	remove vmail xmt annunciator icon
UI_VMAILRCV	lineID		annunc:"RECEIVING VOICE MAIL"
UI_VMAILRCVD		*job	annunc:upd vmail msg count; upd MSG dialog
UI_FILEXFRSTS		*job	DAT:upd file xfr progress bar
UI_FILEXFREND		*job	DAT:upd file xfr progress bar;upd files.dir; remove file xmt annunc icon
UI_FILEXFRABORT		*job	DAT:say"TRANSFER ABORTED" in prog bar;remove file xmt annunc icon
UI_FILEXFRFAIL		*job	DAT:say"COMMUNICATIONS FAILURE" in prog bar; remove file xmt annunc icon
UI_AUDIOSTS		*job	update audio control progress bar
UI_PLAYDONE	lineID		annunc:"ANSWERING MACHINE"
UI_OGMPLAY	lineID		annunc:"PLAYING OUTGOING MESSAGE"
UI_INFOACK		*job	upd directory assistance dialog
UI_INFO		*job	upd directory assistance dialog: update progress bar
UI_INFOEND		*job	upd directory assistance dialog
UI_INFOFAIL		*job	upd directory assistance dialog: say"~~COMMUNICATIONS FAILURE~~ Directory Assistance not Available" in prog bar

INFO

UI_START #records

UI_INFOREC direct

UI_INFOFAIL

NOTE: if job.state = JS_DONE, the UI must remove the job after the action is performed!

Audio Engine Logic Flow

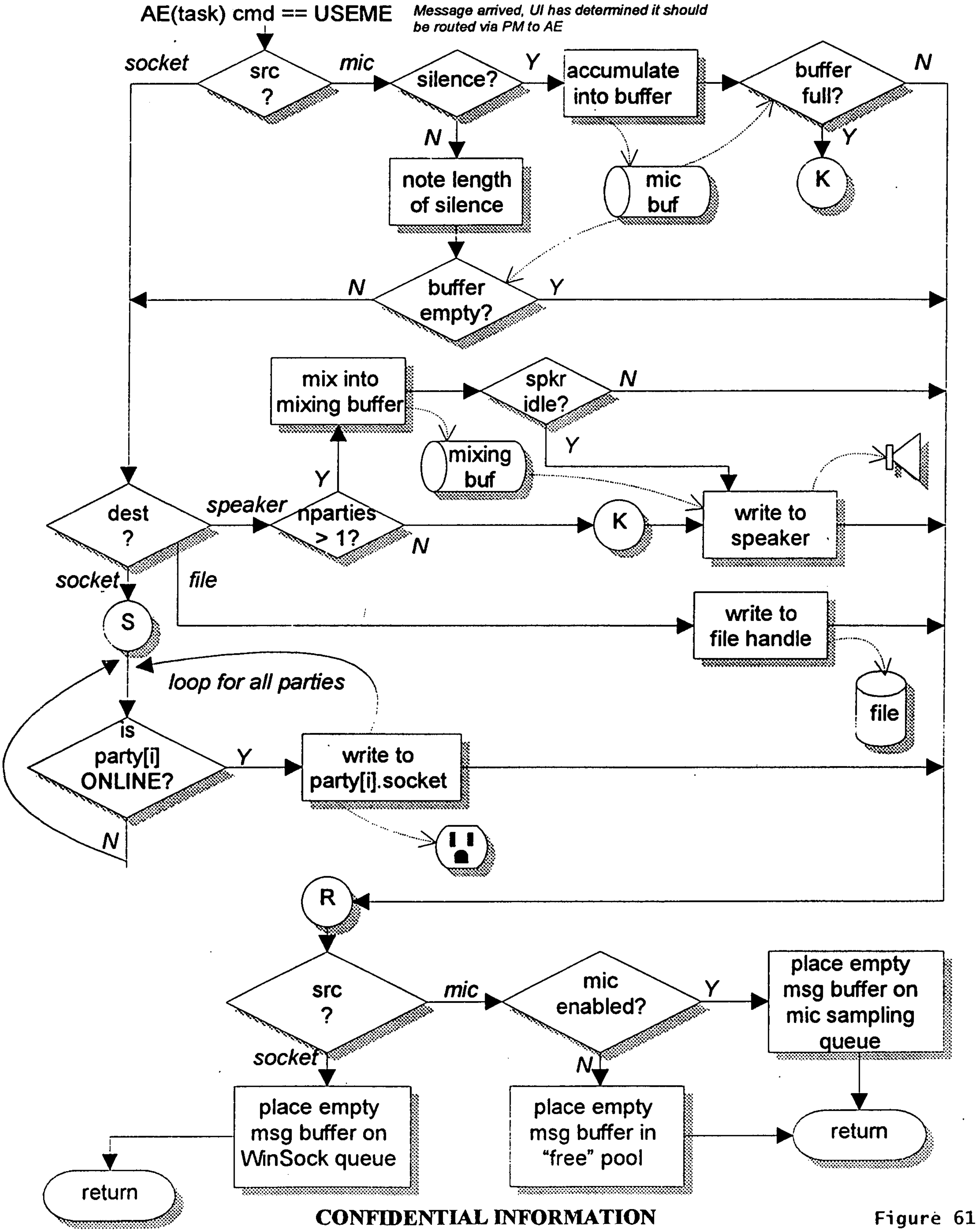
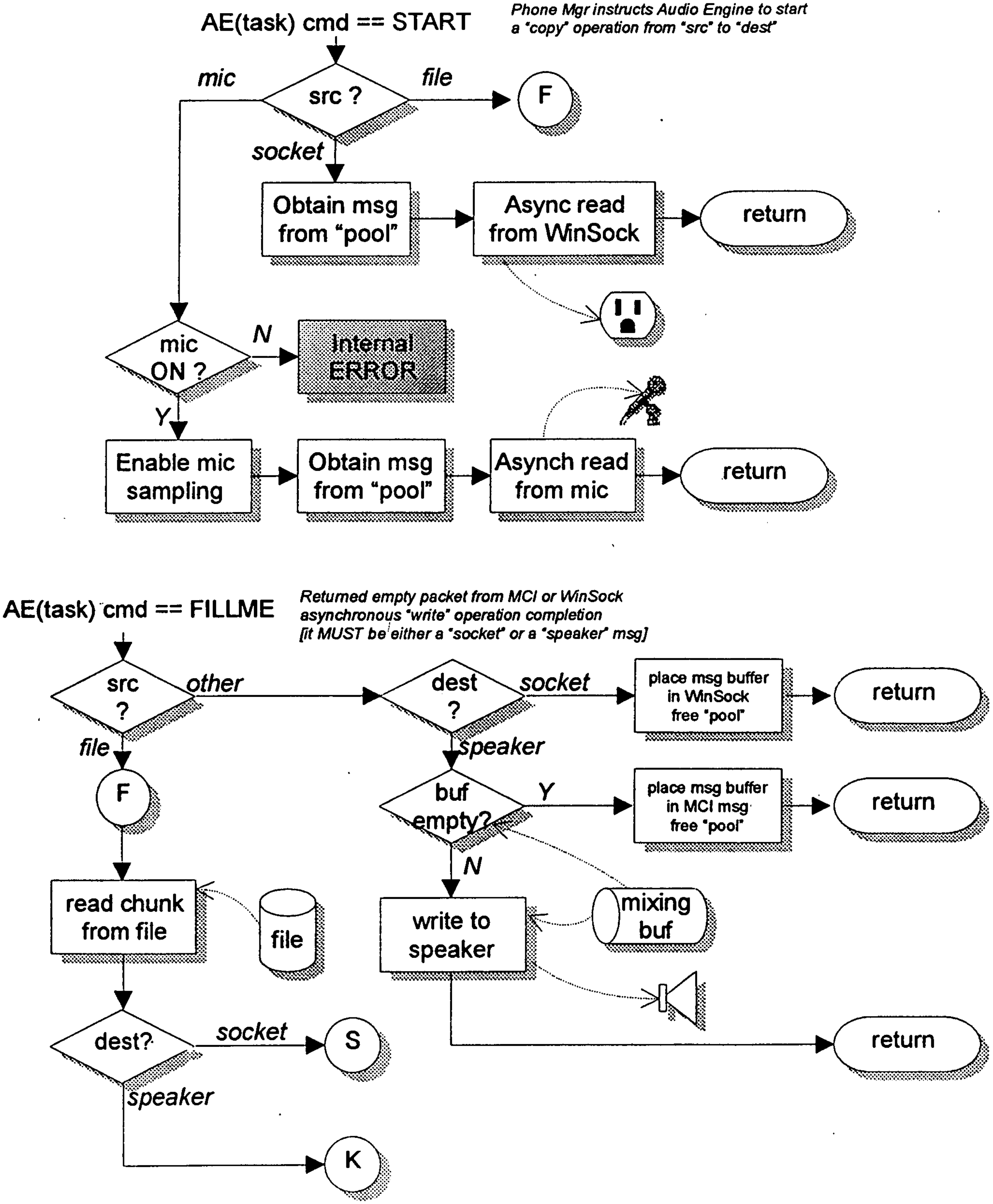


Figure 61

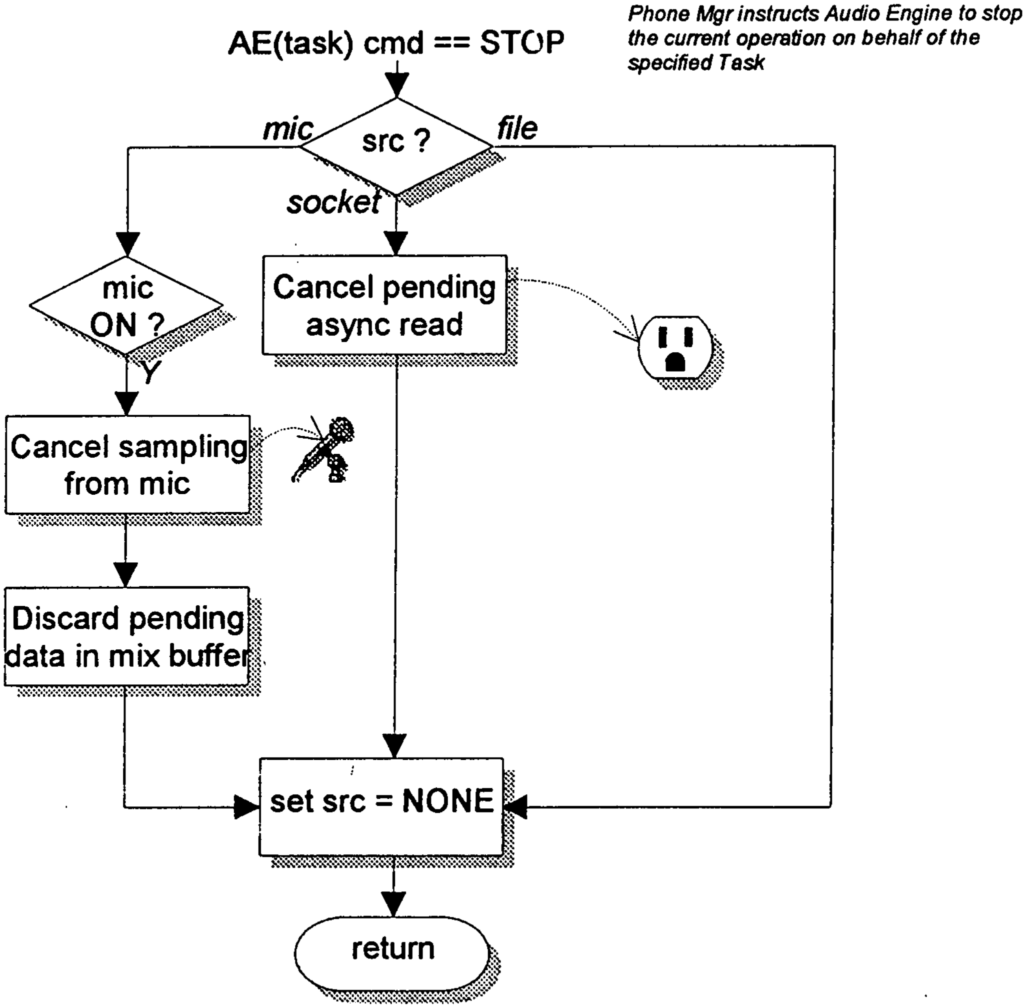
Audio Engine Logic Flow



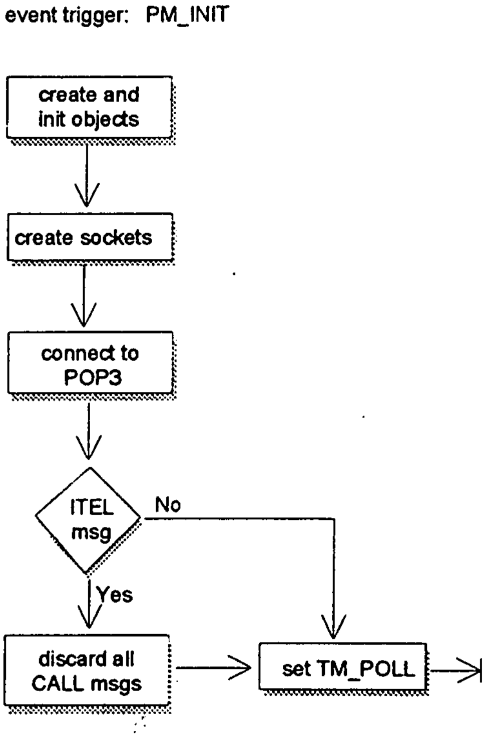
Phone Mgr instructs Audio Engine to start a "copy" operation from "src" to "dest"

Returned empty packet from MCI or WinSock asynchronous "write" operation completion [it MUST be either a "socket" or a "speaker" msg]

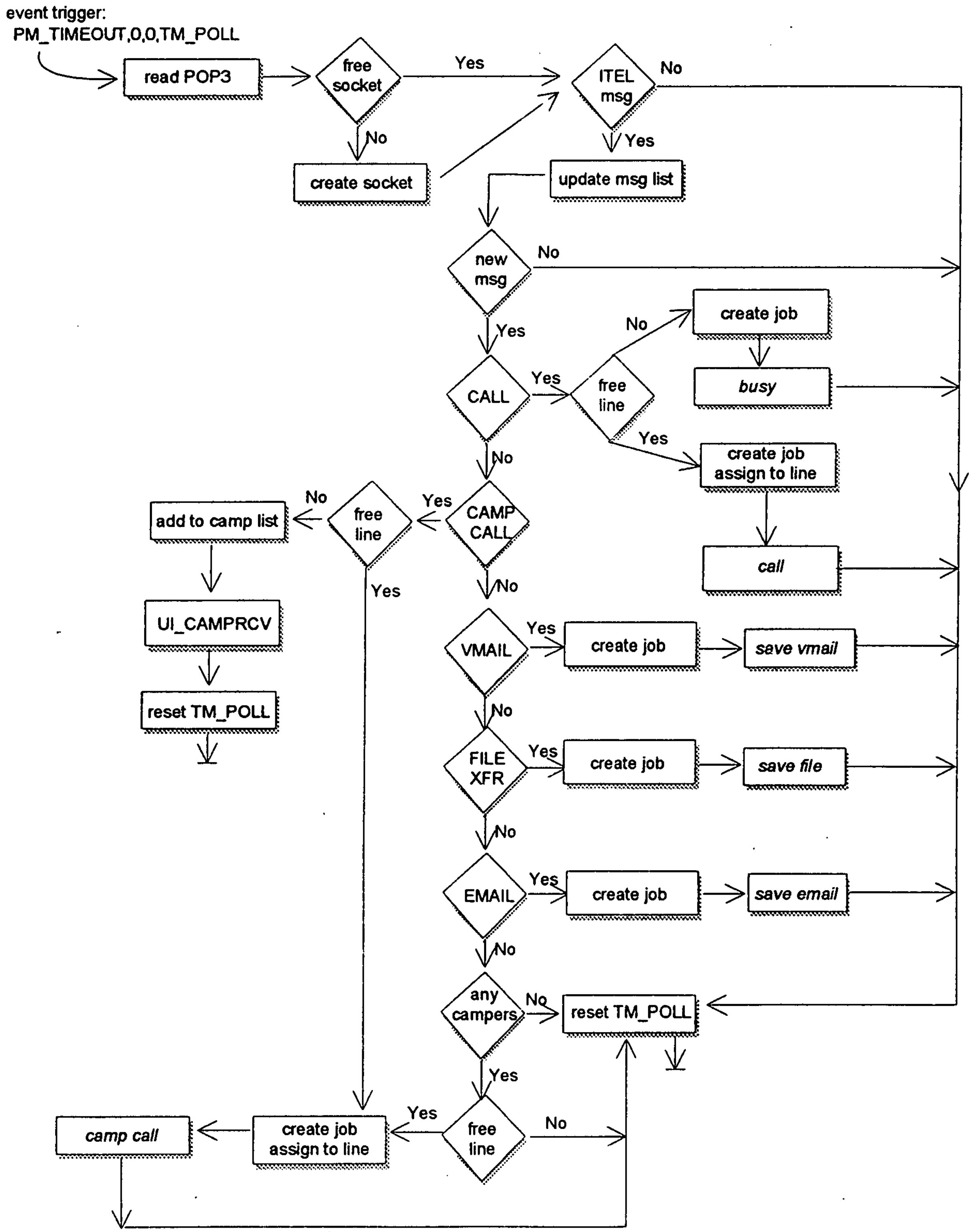
Audio Engine Logic Flow


AE(task) cmd == STOP
Phone Mgr instructs Audio Engine to stop the current operation on behalf of the specified Task
src ?
mic
file
socket
mic ON ?
Y
Cancel pending async read
Cancel sampling from mic
Discard pending data in mix buffer
set src = NONE
return

PhoneManager Init Function



PhoneManager Polling Function


event trigger:
PM_TIMEOUT,0,0,TM_POLL
read POP3
free socket
Yes
No
create socket
ITEL msg
No
Yes
update msg list
new msg
No
Yes
CALL
Yes
free line
No
create job
busy
Yes
create job assign to line
call
No
CAMP CALL
Yes
free line
No
add to camp list
UI_CAMPRCV
reset TM_POLL
Yes
No
VMAIL
Yes
create job
save vmail
No
FILE XFR
Yes
create job
save file
No
EMAIL
Yes
create job
save email
No
any campers
No
reset TM_POLL
Yes
free line
Yes
No
create job assign to line
camp call